Proceedings of Machine Learning Research vol 65:1–37, 2017

Memory and Communication Efficient Distributed Stochastic Optimization with Minibatch-Prox

Jialei Wang[*] JIALEI@UCHICAGO.EDU
University of Chicago

Weiran Wang[*] WEIRANWANG@TTIC.EDU
Toyota Technological Institute at Chicago

Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago

Abstract

We present and analyze an approach for distributed stochastic optimization which is statistically optimal and achieves near-linear speedups (up to logarithmic factors). Our approach allows a communication-memory tradeoff, with either logarithmic communication but linear memory, or polynomial communication and a corresponding polynomial reduction in required memory. This communication-memory tradeoff is achieved through minibatch-prox iterations (minibatch passive-aggressive updates), where a subproblem on a minibatch is solved at each iteration. We provide a novel analysis for such a minibatch-prox procedure which achieves the statistical optimal rate regardless of minibatch size and smoothness, thus significantly improving on prior work.

1. Introduction

Consider the stochastic convex optimization (generalized learning) problem (Nemirovskii and Yudin, 1983; Vapnik, 1995; Shalev-Shwartz et al., 2009):

$$\min_{\mathbf{w}\in\Omega} \phi(\mathbf{w}) := \mathbb{E}_{\xi\sim D}\left[\ell(\mathbf{w},\xi)\right] \tag{1}$$

where our goal is to learn a predictor $\mathbf{w}$ from the convex domain Ω given the convex instantaneous (loss) function $\ell(\mathbf{w},\xi)$ and i.i.d. samples $\xi_1,\xi_2,\dots$ from some unknown data distribution D. When optimizing on a single machine, stochastic approximation methods such as stochastic gradient descent (SGD) or more generally stochastic mirror descent, are ideally suited for the problem as they typically have optimal sample complexity requirements, and run in linear time in the number of samples, and thus also have optimal runtime. Focusing on an ℓ_2 bounded domain with $B = \sup_{\mathbf{w}\in\Omega}\|\mathbf{w}\|$ and L-Lipschitz loss, the min-max optimal sample complexity is $n(\varepsilon) = \mathcal{O}(L^2B^2/\varepsilon^2)$, and this is achieved by SGD using $\mathcal{O}(n(\epsilon))$ vector operations. Furthermore, if examples are obtained one at a time (in a streaming setting or through access to a "button" generating examples), we only need to store $\mathcal{O}(1)$ vectors in memory.

The situation is more complex in the distributed setting where no single method is known that is optimal with respect to sample complexity, runtime, memory *and* communication. Specifically, consider m machines where each machine $i = 1,\dots,m$ receives samples $\xi_{i1},\xi_{i2},\dots$ drawn from the

[*] Equal contributions.

same distribution D. This can equivalently be thought of as randomly distributing samples across m servers. We also assume the objective is β-smooth, taking $L, \beta = \mathcal{O}(1)$ in our presentation of results. The goal is to find a predictor $\hat{\mathbf{w}} \in \Omega$ satisfying $\mathbb{E}\left[\phi(\hat{\mathbf{w}}) - \min_{\mathbf{w}\in\Omega} \phi(\mathbf{w})\right] \le \varepsilon$ using *the smallest possible number of samples* per machine, *the minimal elapsed runtime*, and *the smallest amount of communication*, and also *minimal memory* on each machine (again, when examples are received or generated one at a time). Ideally, we could hope for a method with linear speedup, i.e. $\mathcal{O}(n(\epsilon)/m)$ runtime, using the statistically optimal number of samples $\mathcal{O}(n(\epsilon))$ and constant or near-constant communication and memory. Throughout we measure runtime in terms of vector operations, memory in terms of number of vectors that need to be stored on each machine and communication in terms of number of vectors sent per machine[1]. These resource requirements are summarized in Table 1.

One simple approach for distributed stochastic optimization is minibatch SGD (Cotter et al., 2011; Dekel et al., 2012), where in each update we use a gradient estimate based on mb examples: b examples from each of the m machines. Distributed minibatch SGD attains optimal statistical performance with $\mathcal{O}\left(n(\varepsilon)/m\right)$ runtime, as long as the minibatch size is not too large: Dekel et al. (2012) showed that the minibatch size can be as large as $bm = \mathcal{O}(\sqrt{n(\varepsilon)})$, and Cotter et al. (2011) showed that with acceleration this can be increased to $bm = \mathcal{O}(n(\varepsilon)^{3/4})$. Using this maximal minibatch size for accelerated minibatch SGD thus yields a statistically optimal method with linear speedup in runtime, $\mathcal{O}(1)$ memory usage, and $\mathcal{O}(n(\varepsilon)^{1/4})$ rounds of communication–see Table 1. This is the most communication-efficient method with true linear speedup we are aware of.

An alternative approach is to use distributed optimization to optimize the regularized empirical objective:

$$\min_{\mathbf{w}} \ \phi_S(\mathbf{w}) + \frac{\nu}{2} \left\|\mathbf{w}\right\|^2, \tag{2}$$

where ϕ_S is the empirical objective on $n(\epsilon)$ i.i.d. samples, distributed across the machines and $\nu = \mathcal{O}(L/(B\sqrt{n(\varepsilon)}))$. A naive approach here is to use accelerate gradient descent, distributing the gradient computations, but this, as well as approaches based on ADMM (Boyd et al., 2011), are dominated by minibatch SGD (Shamir and Srebro 2014 and see also Table 1). Better alternatives take advantage of the stochastic nature of the problem: DANE (Shamir et al., 2014) requires only $\mathcal{O}(B^2 m)$ rounds of communication for squared loss problems, while DiSCO (Zhang and Lin, 2015) and AIDE (Reddi et al., 2016)) reduce this further to $\mathcal{O}(B^{1/2} m^{1/4})$ rounds of communication. However, these communication-efficient methods usually require expensive computation on each local machine, solving an optimization problem on all local data at each iteration. Even if this can be done in near-linear time, it is still difficult to obtain computational speedup compared with single machine solution, and certainly not linear speedups—see Table 1. Furthermore, since each round of these methods involves optimization over a fixed training set, this training set must be stored thus requiring $n(\varepsilon)/m$ memory per machine.

Designing stochastic distributed optimization problems with linear, or near-linear, speedups, and low communication and memory requirements is thus still an open problem. We make progress in this paper analyzing and presenting methods with near-linear speedups and better communication and memory requirements. As with the analysis of DANE, DiSCO and AIDE, our analysis is rigorous only for least squared problems, and so all results should be taken in that context (the methods themselves are applicable to any distributed stochastic convex optimization problem).

1. In all methods involved, communication is used to average vectors across machines and make the result known to one or all machines. We are actually counting the number of such operations.

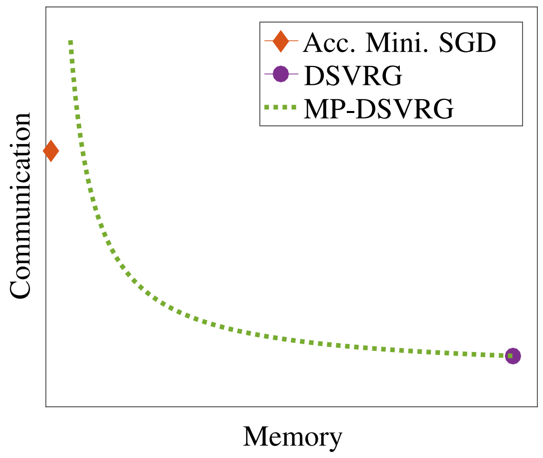

Figure 1: Trade-offs between memory and communication for the proposed MP-DSVRG approach.

Our contributions

- We first apply the recently proposed distributed SVRG (DSVRG) algorithm for regularized loss minimization to the distributed stochastic convex optimization problem, and show that on least square problems it can achieve near-linear speedup with very low communication, but with high memory cost—see DSVRG in Table 1.

- We propose a novel algorithm that improves the memory cost, which we call minibatch-prox with DSVRG (MP-DSVRG). For least square problems it achieves near-linear speedup with communication cost that is higher than DSVRG but increases only logarithmically with $n(\varepsilon)$, but with much lower memory requirements. Moreover, our algorithm is flexible, allowing to trade off between communication and memory (depicted in Figure 1), without affecting the computational efficiency. Our method is based on careful combinations of inexact minibatch proximal update, communication-efficient optimization and linearly convergent stochastic gradient algorithms for finite-sums.

- As indicated above, our method is based on minibatch proximal update. That is, a minibatch approach where in each iteration a non-linearized problem is solved on a stochastic minibatch. This can be viewed as a minibatch generalization to the passive-aggressive algorithm (Crammer et al., 2006) and has been considered in various contexts (Kulis and Bartlett, 2010; Toulis and Airoldi, 2014). We show that such an approach achieves the optimal statistical rate in terms of the number of samples used independent of the number of iterations, i.e. with *any* minibatch size. This significantly improves over the previous analysis of Li et al. (2014), as the guarantee is better, it entirely avoid the dependence on the minibatch size and does not rely on additional assumptions as in Li et al. (2014). The guarantee holds for any Lipschitz (even non-smooth) objective. Furthermore, to make the minibatch proximal iterate more practical and useful in distributed setting, we also extend the analysis to algorithms which solve each minibatch subproblem inexactly. Our analysis of exact and inexact mini-

	Samples	Communication	Computation	Memory
Ideal Solution	$n(\varepsilon)$	$\mathcal{O}(1)$	$n(\varepsilon)/m$	$\mathcal{O}(1)$
Accelerated GD	$n(\varepsilon)$	$B^{1/2}n(\varepsilon)^{1/4}$	$B^{1/2}n(\varepsilon)^{5/4}/m$	$n(\varepsilon)/m$
Acc. Minibatch SGD	$n(\varepsilon)$	$B^{1/2}n(\varepsilon)^{1/4}$	$n(\varepsilon)/m$	$\mathcal{O}(1)$
DANE	$n(\varepsilon)$	$B^2 m$	$B^2 n(\varepsilon)$	$n(\varepsilon)/m$
DiSCO	$n(\varepsilon)$	$B^{1/2}m^{1/4}$	$B^{1/2}n(\varepsilon)/m^{3/4}$	$n(\varepsilon)/m$
AIDE	$n(\varepsilon)$	$B^{1/2}m^{1/4}$	$B^{1/2}n(\varepsilon)/m^{3/4}$	$n(\varepsilon)/m$
DSVRG	$n(\varepsilon)$	$\mathcal{O}(1)$	$n(\varepsilon)/m$	$n(\varepsilon)/m$
MP-DSVRG ($b \le b_{\max}$)	$n(\varepsilon)$	$n(\varepsilon)/(mb)$	$n(\varepsilon)/m$	b
MP-DSVRG ($b = b_{\max}$)	$n(\varepsilon)$	$\mathcal{O}(1)$	$n(\varepsilon)/m$	$n(\varepsilon)/m$

Table 1: Summary of resources required by different approaches to distributed stochastic least squares problems, in units of vector operations/communications/memory per machine, ignoring constants and log-factors, here $b_{\max} = n(\varepsilon)/m$.

batch proximal updates may be of independent interest and useful in other contexts and as a basis for other methods.

Notations We denote by $\mathbf{w}_* = \arg\min_{\mathbf{w}\in\Omega} \phi(\mathbf{w})$ the optimal solution to (1). Throughout the paper, we assume the instantaneous function $\ell(\mathbf{w},\xi)$ is L-Lipschitz and λ-strongly convex in $\mathbf{w}$ for some $\lambda \ge 0$ on the domain Ω:

$$\left|\ell(\mathbf{w},\xi) - \ell(\mathbf{w}',\xi)\right| \le L \left\|\mathbf{w} - \mathbf{w}'\right\|,$$
$$\ell(\mathbf{w},\xi) - \ell(\mathbf{w}',\xi) \ge \left\langle \nabla\ell(\mathbf{w}',\xi),\, \mathbf{w} - \mathbf{w}'\right\rangle + \frac{\lambda}{2}\left\|\mathbf{w} - \mathbf{w}'\right\|^2, \qquad \forall \mathbf{w},\mathbf{w}' \in \Omega.$$

Sometimes we also assume $\ell(\mathbf{w},\xi)$ is β-smooth in $\mathbf{w}$:

$$\ell(\mathbf{w},\xi) - \ell(\mathbf{w}',\xi) \le \left\langle \nabla\ell(\mathbf{w}',\xi),\, \mathbf{w} - \mathbf{w}'\right\rangle + \frac{\beta}{2}\left\|\mathbf{w} - \mathbf{w}'\right\|^2, \qquad \forall \mathbf{w},\mathbf{w}' \in \Omega.$$

For distributed stochastic optimization, our analysis focuses on the least squares loss $\ell(\mathbf{w},\xi) = \frac{1}{2}(\mathbf{w}^\top\mathbf{x} - y)^2$ where $\xi = (\mathbf{x}, y)$.

2. Distributed SVRG for stochastic convex optimization

Recently, Lee et al. (2015) suggested using fast randomized optimization algorithms for finite-sums, and in particular the SVRG algorithm, as a distributed optimization approach for (2). The authors noted that, for SVRG, when the the sample size $n(\varepsilon)$ dominates the problem's condition number β/ν where β is the smoothness parameter of $\ell(\mathbf{w},\xi)$, the time complexity is dominated by computing the batch gradients. This operation can be trivially parallelized. The stochastic updates, on the other hand, can be implemented on a single machine while the other machines wait, with the only caveat being that only sampling-without-replacement can be implemented this way. The use of without-replacement sampling was theoretically justified in a recent analysis by Shamir (2016).

In the distributed stochastic convex optimization setting considered here, DSVRG in fact achieves linear speedup in certain regime as follows. In each iteration of the algorithm, each machine first computes its local gradient and average them with one communication round to obtain the global batch gradient, and then a *single* machine performs the SVRG stochastic updates by processing its local data once (sampling the $n(\varepsilon)/m$ examples without replacement). By the linear convergence of SVRG, as long as the number of stochastic updates $n(\varepsilon)/m$ is larger than $\beta/\nu = \mathcal{O}(\beta B\sqrt{n(\varepsilon)}/L)$, the algorithm converges to $\mathcal{O}(\epsilon)$-suboptimality (in both the empirical and stochastic objective) in $\mathcal{O}(\log 1/\varepsilon) = \mathcal{O}\left(\log n(\varepsilon)\right)$ iterations; and this condition is satisfied[2] for $n(\varepsilon) \gtrsim m^2$.

Clearly, in the above regime, each iteration of DSVRG uses two rounds of communications and the total communication complexity is $\mathcal{O}\left(n(\varepsilon)\right)$. On the other hand, the computation for each machine is compute the local gradient (in time $\mathcal{O}(n(\varepsilon)/m)$) in each iteration, resulting in a total time complexity of $\mathcal{O}(n(\varepsilon)\log n(\varepsilon)/m)$. This explains the DSVRG entry in Table 1.

Being communication- and computation-efficient, DSVRG requires each machine to store a portion of the sample set for ERM to make multiple passes over them, and is therefore not memory-efficient. In fact, this disadvantage is shared by previously known communication-efficient distributed optimization algorithms, including DANE, DiSCO, and AIDE. In order to develop a memory- and communication-efficient algorithm for distributed stochastic optimization, we need to bypass the ERM setting and this is enabled by the following minibatch-prox algorithm.

3. The minibatch-prox algorithm for stochastic optimization

In this section, we describe and analyze the minibatch-prox algorithm for stochastic optimization, which allows us to use arbitrarily large minibatch size without slowing down the convergence rate. We first present the basic version where each proximal objective is solved exactly for each minibatch, which achieves the optimal convergence rate. Then, we show that if each minibatch objective is solved accurately enough, the algorithm still converges at the optimal rate, opening the opportunity for efficient implementations.

3.1. Exact minibatch-prox

The "exact" minibatch-prox is defined by the following iterates: for $t = 1, \dots,$

$$\mathbf{w}_t = \operatorname*{arg\,min}_{\mathbf{w}\in\Omega}\ f_t(\mathbf{w}),$$
$$\text{where} \quad f_t(\mathbf{w}) := \phi_{I_t}(\mathbf{w}) + \frac{\gamma_t}{2}\left\|\mathbf{w}-\mathbf{w}_{t-1}\right\|^2 = \frac{1}{b}\sum_{\xi\in I_t}\ell(\mathbf{w},\xi) + \frac{\gamma_t}{2}\left\|\mathbf{w}-\mathbf{w}_{t-1}\right\|^2, \tag{3}$$

$\gamma_t > 0$ is the (inverse) stepsize parameter at time t, and I_t is a set of a b samples from the unknown distribution D. To understand the updates in (3), we first observe by the first order optimality condition for $f_t(\mathbf{w})$ that

$$\nabla\phi_{I_t}(\mathbf{w}_t) + \gamma_t(\mathbf{w}_t - \mathbf{w}_{t-1}) \in -\mathcal{N}_\Omega(\mathbf{w}_t), \tag{4}$$

2. If $n(\varepsilon) \gtrsim m^2$ does not hold, we can use a "hot-potato" style algorithm where we process all data once on machine i and pass the predictor to machine $i+1$ until we obtain sufficiently many stochastic updates. But then the computation efficiency deteriorates and we no longer have linear speedup in runtime.

where $\nabla\phi_{I_t}(\mathbf{w}_t)$ is some subgradient of $\phi_{I_t}(\mathbf{w})$ at $\mathbf{w}_t$, and $\mathcal{N}_\Omega(\mathbf{w}_t) = \{\mathbf{y} | \langle \mathbf{w} - \mathbf{w}_t,\, \mathbf{y}\rangle \le 0,\, \forall \mathbf{w} \in \Omega\}$ is the normal cone of Ω at $\mathbf{w}_t$. Equivalently, the above condition implies

$$\mathbf{w}_t = P_\Omega\left(\mathbf{w}_{t-1} - \frac{1}{\gamma_t}\nabla\phi_{I_t}(\mathbf{w}_t)\right), \tag{5}$$

where $P_\Omega(\mathbf{w})$ denotes the projection of $\mathbf{w}$ onto Ω. The update rule (5) resembles that of the standard minibatch gradient descent, except the gradient is evaluated at the "future" iterate.

Proximal steps, of the form (3) or equivalently (5), are trickier to implement compared to (stochastic) gradient steps, as they involve optimization of a subproblem, instead of merely computing and adding gradients. Nevertheless, they have been suggested, used and studied in several contexts. Crammer et al. (2006) proposed the "passive aggressive" update rule, where a margin-based loss from a single example with a quadratic penalty is minimized—this corresponds to (3) with a "batch size" of one. More general loss functions, still for "batch sizes" of one, were also analyzed in the online learning setting (Cheng et al., 2006; Kulis and Bartlett, 2010). For finite-sum objectives, methods based on incremental/stochastic proximal updates were studied by Bertsekas (2011, 2015); Defazio (2016). Needell and Tropp (2014) analyzed a randomized block Kaczmarz method in the context of solving linear systems, which also minimizes the empirical loss on a randomly sampled minibatch. To the best of our knowledge, no prior work has analyzed the general minibatch variant of proximal updates for stochastic optimization except Li et al. (2014). However, the analysis of Li et al. (2014) assumes a stringent condition which is hard to verify (and is often violated) in practice, which we will discuss in detail in this section.

The following lemma provides the basic property of the update at each iteration.

Lemma 1 *For any $\mathbf{w} \in \Omega$, we have*

$$\frac{\lambda + \gamma_t}{\gamma_t}\left\|\mathbf{w}_t - \mathbf{w}\right\|^2 \le \left\|\mathbf{w}_{t-1} - \mathbf{w}\right\|^2 - \left\|\mathbf{w}_{t-1} - \mathbf{w}_t\right\|^2 - \frac{2}{\gamma_t}\left(\phi_{I_t}(\mathbf{w}_t) - \phi_{I_t}(\mathbf{w})\right). \tag{6}$$

To derive the convergence guarantee, we need to relate $\phi_{I_t}(\mathbf{w}_t)$ to $\phi(\mathbf{w})$. The analysis of Li et al. (2014) for minibatch-prox made the assumption that for all $t \ge 1$:

$$\mathbb{E}_{I_t}\left[D_\phi(\mathbf{w}_t; \mathbf{w}_{t-1})\right] \le \mathbb{E}_{I_t}\left[D_{\phi_{I_t}}(\mathbf{w}_t; \mathbf{w}_{t-1})\right] + \frac{\gamma_t}{2}\left\|\mathbf{w}_t - \mathbf{w}_{t-1}\right\|^2, \tag{7}$$

where $D_f(\mathbf{w}, \mathbf{w}') = f(\mathbf{w}) - f(\mathbf{w}') - \langle \nabla f(\mathbf{w}'),\, \mathbf{w} - \mathbf{w}'\rangle$ denotes the Bregman divergence defined by the potential function f. This condition is hard to verify, and may constrain the stepsize to be very small. For example, as the authors argued, if $\ell(\mathbf{w}, \xi)$ is β-smooth with respect to $\mathbf{w}$, we have

$$D_\phi(\mathbf{w}_t; \mathbf{w}_{t-1}) \le \frac{\beta}{2}\left\|\mathbf{w}_t - \mathbf{w}_{t-1}\right\|^2,$$

and combined with the fact that $D_{\phi_{I_t}}(\mathbf{w}_t; \mathbf{w}_{t-1}) \ge 0$, one can guarantee (7) by setting $\gamma_t \ge \beta$. However, to obtain the optimal convergence rate, Li et al. (2014) needed to set $\gamma_t = \mathcal{O}(\sqrt{T/b})$ which would imply $b = \mathcal{O}(T)$ in order to have $\gamma_t \ge \beta$. In view of this implicit constraint that the minibatch size b can not be too large, the analysis of Li et al. (2014) does not really show advantage of minibatch-prox over minibatch SGD, whose optimal minibatch size is precisely $b = \mathcal{O}(T)$.

Our analysis is free of any additional assumptions. The key observation is that, when b is large, we expect $\phi_{I_t}(\mathbf{w})$ to be close to $\phi(\mathbf{w})$. Define the stochastic objective

$$F_t(\mathbf{w}) := \mathbb{E}_{I_t}\left[f_t(\mathbf{w})\right] = \phi(\mathbf{w}) + \frac{\gamma_t}{2}\left\|\mathbf{w} - \mathbf{w}_{t-1}\right\|^2. \tag{8}$$

Then $\mathbf{w}_t$ is the "empirical risk minimizer" of $F_t(\mathbf{w})$ as it solves the empirical version $f_t(\mathbf{w})$ with b samples. Using a stability argument (Shalev-Shwartz et al., 2009), we can establish the "generalization" performance for the (inexact) minimizer of the minibatch objective.

Lemma 2 *For the minibatch-prox algorithm,we have*

$$\left|\mathbb{E}_{I_t}\left[\phi(\mathbf{w}_t) - \phi_{I_t}(\mathbf{w}_t)\right]\right| \le \frac{4L^2}{(\lambda+\gamma_t)b}.$$

Moreover, if a possibly randomized algorithm $\mathcal{A}$ minimizes $f_t(\mathbf{w})$ up to an error of η_t, i.e., $\mathcal{A}$ returns an approximate solution $\tilde{\mathbf{w}}_t$ such that $\mathbb{E}_{\mathcal{A}}\left[f_t(\tilde{\mathbf{w}}_t) - f_t(\mathbf{w}_t)\right] \le \eta_t$, we have

$$\left|\mathbb{E}_{I_t,\mathcal{A}}\left[\phi(\tilde{\mathbf{w}}_t) - \phi_{I_t}(\mathbf{w}_t)\right]\right| \le \frac{4L^2}{(\lambda+\gamma_t)b} + \sqrt{\frac{2L^2\eta_t}{\lambda+\gamma_t}}.$$

Combining Lemma 1 and Lemma 2, we obtain the following key lemma regarding the progress on the stochastic objective at each iteration of minibatch-prox.

Lemma 3 *For iteration t of exact minibatch-prox, we have for any $\mathbf{w} \in \Omega$ that*

$$\frac{\lambda+\gamma_t}{\gamma_t}\mathbb{E}_{I_t}\left\|\mathbf{w}_t - \mathbf{w}\right\|^2 \le \left\|\mathbf{w}_{t-1} - \mathbf{w}\right\|^2 - \frac{2}{\gamma_t}\mathbb{E}_{I_t}\left[\phi(\mathbf{w}_t) - \phi(\mathbf{w})\right] + \frac{8L^2}{\gamma_t(\lambda+\gamma_t)b}. \tag{9}$$

We are now ready to bound the overall convergence rates of minibatch-prox.

Theorem 4 (Convergence of exact minibatch-prox — weakly convex $\ell(\mathbf{w},\xi)$) *For L-Lipschitz instantaneous function $\ell(\mathbf{w},\xi)$, set $\gamma = \sqrt{\frac{8T}{b}} \cdot \frac{L}{\|\mathbf{w}_0 - \mathbf{w}_*\|}$ for $t = 1, \dots, T$ in minibatch-prox. Then for $\widehat{\mathbf{w}}_T = \frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t$, we have*

$$\mathbb{E}\left[\phi(\widehat{\mathbf{w}}_T) - \phi(\mathbf{w}_*)\right] \le \frac{\sqrt{8}L}{\sqrt{bT}}\left\|\mathbf{w}_0 - \mathbf{w}_*\right\|.$$

Theorem 5 (Convergence of exact minibatch-prox — strongly convex $\ell(\mathbf{w},\xi)$) *For L-Lipschitz and λ-strongly convex instantaneous function $\ell(\mathbf{w},\xi)$, set $\gamma_t = \frac{\lambda(t-1)}{2}$ for $t = 1, \dots, T$ in minibatch-prox. Then for $\widehat{\mathbf{w}}_T = \frac{2}{T(T+1)}\sum_{t=1}^{T} t\mathbf{w}_t$, we have*

$$\mathbb{E}\left[\phi(\widehat{\mathbf{w}}_T) - \phi(\mathbf{w}_*)\right] \le \frac{16L^2}{\lambda b(T+1)}.$$

3.2. Inexact minibatch-prox

We now study the case where instead of solving the subproblems $f_t(\mathbf{w})$ exactly, we only solve it approximately to sufficient accuracy. The "inexact" minibatch-prox uses a possibly randomized algorithm $\mathcal{A}$ for approximately solving one subproblem on a minibatch in each iteration, and generates the following iterates: for $t = 1, \dots,$

$$\tilde{\mathbf{w}}_t \approx \bar{\mathbf{w}}_t := \arg\min_{\mathbf{w}\in\Omega} \ \tilde{f}_t(\mathbf{w}) \qquad \text{where} \quad \tilde{f}_t(\mathbf{w}) := \phi_{I_t}(\mathbf{w}) + \frac{\gamma_t}{2}\left\|\mathbf{w} - \tilde{\mathbf{w}}_{t-1}\right\|^2, \tag{10}$$
$$\text{and} \qquad \mathbb{E}_{\mathcal{A}}\left[\tilde{f}_t(\tilde{\mathbf{w}}_t) - \tilde{f}_t(\bar{\mathbf{w}}_t)\right] \le \eta_t.$$

Analogous to Lemma 3, we can derive the following lemma using stability of inexact minimizers.

Lemma 6 *Fix any $\mathbf{w} \in \Omega$. For iteration t of inexact minibatch-prox, we have*

$$\mathbb{E}_{I_t,\mathcal{A}}\left[\phi(\tilde{\mathbf{w}}_t) - \phi(\mathbf{w})\right] \le \frac{\gamma_t}{2}\mathbb{E}_{I_t,\mathcal{A}}\left\|\tilde{\mathbf{w}}_{t-1} - \mathbf{w}\right\|^2 - \frac{\lambda+\gamma_t}{2}\mathbb{E}_{I_t,\mathcal{A}}\left\|\tilde{\mathbf{w}}_t - \mathbf{w}\right\|^2 + \frac{4L^2}{(\lambda+\gamma_t)b}$$
$$+ \sqrt{\frac{2L^2\eta_t}{\lambda+\gamma_t}} + \sqrt{2(\lambda+\gamma_t)\eta_t}\cdot\sqrt{\mathbb{E}_{I_t,\mathcal{A}}\left\|\tilde{\mathbf{w}}_t - \mathbf{w}\right\|^2}. \tag{11}$$

Note that when $\eta_t = 0$, the above guarantee reduces to that of exact minibatch-prox.

We now show that when the minibatch subproblems are solved sufficiently accurately, we still obtain the $\mathcal{O}(1/\sqrt{bT})$ rate for weakly-convex loss and $\mathcal{O}(1/(\lambda bT))$ rate for strongly-convex loss.

Theorem 7 (Convergence of inexact minibatch-prox — weakly convex $\ell(\mathbf{w},\xi)$) *For L-Lipschitz instantaneous function $\ell(\mathbf{w},\xi)$, set $\gamma_t = \gamma = \sqrt{\frac{8T}{b}}\cdot\frac{L}{\|\mathbf{w}_0-\mathbf{w}_*\|}$ for all $t \ge 1$ in inexact minibatch-prox. Assume that for all $t \ge 1$, the error in minimizing $\tilde{f}_t(\mathbf{w})$ satisfies for some $\delta > 0$ that*

$$\mathbb{E}_{\mathcal{A}}\left[\tilde{f}_t(\tilde{\mathbf{w}}_t) - \min_{\mathbf{w}}\tilde{f}_t(\mathbf{w})\right] \le \min\left(c_1\left(\frac{T}{b}\right)^{\frac{1}{2}}, c_2\left(\frac{T}{b}\right)^{\frac{3}{2}}\right)\cdot\frac{L\left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|}{t^{2+2\delta}}.$$

Then for $\widehat{\mathbf{w}}_T = \frac{1}{T}\sum_{t=1}^T \tilde{\mathbf{w}}_t$, we have $\mathbb{E}\left[\phi(\widehat{\mathbf{w}}_T) - \phi(\mathbf{w}_*)\right] \le \frac{c_3 L\|\mathbf{w}_0-\mathbf{w}_*\|}{\sqrt{bT}}$, where c_3 only depends on c_1, c_2 and δ. For example, by setting $c_1 = 10^{-4}, c_2 = 10^{-4}, \delta = 1/2$, we have

$$\mathbb{E}\left[\phi(\widehat{\mathbf{w}}_T) - \phi(\mathbf{w}_*)\right] \le \frac{\sqrt{10}L\left\|\mathbf{w}_0 - \mathbf{w}_*\right\|}{\sqrt{bT}}.$$

Theorem 8 (Convergence of inexact minibatch-prox — strongly convex $\ell(\mathbf{w},\xi)$) *For L-Lipschitz and λ-strongly convex instantaneous function $\ell(\mathbf{w},\xi)$, set $\gamma_t = \frac{\lambda(t-1)}{2}$ for $t = 1, \dots$ in inexact minibatch-prox. Assume that for all $t \ge 1$, the error in minimizing $\tilde{f}_t(\mathbf{w})$ satisfies for some $\delta > 0$ that*

$$\mathbb{E}_{\mathcal{A}}\left[\tilde{f}_t(\tilde{\mathbf{w}}_t) - \min_{\mathbf{w}}\tilde{f}_t(\mathbf{w})\right] \le \min\left(c_1\left(\frac{T}{b}\right), c_2\left(\frac{T}{b}\right)^2\right)\cdot\frac{L^2}{t^{3+2\delta}\lambda}.$$

Then for $\widehat{\mathbf{w}}_T = \frac{2}{T(T+1)}\sum_{t=1}^T t\tilde{\mathbf{w}}_t$, we have $\mathbb{E}\left[\phi(\widehat{\mathbf{w}}_T) - \phi(\mathbf{w}_*)\right] \le \frac{c_3 L^2}{\lambda bT}$, where c_3 only depends on c_1, c_2 and δ.

Remark 9 *The final inequalities in Theorem 4 and 7 actually apply more generally to all predictors in the domain. That is, our proofs still hold with $\mathbf{w}^*$ replaced by any $\mathbf{w} \in \Omega$:*

$$\mathbb{E}\left[\phi(\widehat{\mathbf{w}}_T) - \phi(\mathbf{w})\right] \le \mathcal{O}\left(\frac{L\left\|\mathbf{w}_0 - \mathbf{w}\right\|}{\sqrt{bT}}\right), \qquad \mathbf{w} \in \Omega.$$

This allows us to compete with any predictor in the domain (other than the minimizer). For example, in order to compete on $\phi(\mathbf{w})$ with the set of predictors with small norm $\{\mathbf{w} : \|\mathbf{w}\| \le B\}$, we can set the domain $\Omega = \mathbb{R}^d$ and initialize with $\mathbf{w}_0 = \mathbf{0}$. In view of the above inequality, we still obtain the optimal rate $\mathcal{O}\left(\frac{LB}{\sqrt{bT}}\right)$ from minibatch-prox by solving simpler, unconstrained subproblems (though we might have $\|\hat{\mathbf{w}}_T\| > B$).

4. Communication-efficient distributed minibatch-prox with SVRG

We now apply the theoretical results of minibatch-prox to the distributed stochastic learning setting, and propose a novel algorithm that is both communication and computation efficient, and being able to explore trade-offs between memory and communication efficiency.

Suppose we have m machines in a distributed environment. For each outer loop of our algorithm, each machine i draws a minibatch $I_t^{(i)}$ of b samples independently from other machines, and denote $I_t = \cup_{i=1}^m I_t^{(i)}$ which contains bm samples. To apply the minibatch-prox algorithm from the previous section, we need to find an approximate solution to the following problem:

$$\min_{\mathbf{w}} \; \tilde{f}_t(\mathbf{w}) := \phi_{I_t}(\mathbf{w}) + \frac{\gamma}{2}\left\|\mathbf{w} - \mathbf{w}_{t-1}\right\|^2. \tag{12}$$

Since the objective (12) involves functions from different machines, we use distributed optimization algorithms for solving it. In Li et al. (2014), the authors proposed a simple algorithm EMSO to approximately solve (12), where each machine first solve its own local objective, i.e.,

$$\mathbf{w}_t^{(i)} = \arg\min_{\mathbf{w}} \; \phi_{I_t^{(i)}} + \frac{\gamma}{2}\left\|\mathbf{w} - \mathbf{w}_{t-1}\right\|^2, \tag{13}$$

and then all machines average their local solutions via one round of communication: $\mathbf{w}_t = \frac{1}{m}\sum_{i=1}^m \mathbf{w}_t^{(i)}$.

We note that this can be considered as the "one-shot-averaging" approach (Zhang et al., 2012) for solving (12). Although this approach was shown to work well empirically, no convergence guarantee for the original stochastic objective (1) was provided by Li et al. (2014). Here we instead use the distributed SVRG (DSVRG) algorithm (Lee et al., 2015; Shamir, 2016) to approximately solve (12), as DSVRG enjoys excellent communication and computation cost when the problem is well conditioned (cf. Table 1).[3]

We detail our algorithm, named MP-DSVRG (minibatch-prox with DSVRG), in Algorithm 1. The algorithm consists of two nested loops, where t, k are iteration counters for minibatch-prox (the outer **for**-loop), and DSVRG (the inner **for**-loop) respectively. In each outer loop, each machine draws a minibatch $I_t^{(i)}$ to form the objective (12), which will be solved approximately by the inner loops. Moreover, each machine splits its local dataset into p_i batches: $I^{(i)} = \cup_{j=1}^{p_i} B_j^{(i)}$. In

3. It is also possible to equip minibatch-prox with other communication-efficient distributed optimization algorithms, for example in Appendix D, we present a minibatch-prox DANE (MP-DANE) algorithm which uses the accelerated DANE method for solving (12).

Algorithm 1 Minibatch-prox with DSVRG for distributed stochastic convex optimization.

```
Initialize $\mathbf{w}_0 = \mathbf{0}$.
for $t = 1, 2, \ldots, T$ do
  % Outer loop performs minibatch-prox.
  Each machine $i$ draws a minibatch $I_t^{(i)}$ of $b$ samples from the underlying data distribution, and
  split $I_t^{(i)}$ to $p_i$ batches of size $b/p_i$: $B_1^{(i)}, B_2^{(i)}, ..., B_{p_i}^{(i)}$
  Initialize $\mathbf{z}_0 \leftarrow \mathbf{w}_{t-1}, \quad \mathbf{x}_0 \leftarrow \mathbf{w}_{t-1}, \quad j \leftarrow 1, \quad s \leftarrow 1$
  for $k = 1, 2, \ldots, K$ do
    1. All machines perform one round of communication to compute the average gradient:
       $$\nabla\phi_{I_t}(\mathbf{z}_{k-1}) \leftarrow \frac{1}{m}\sum_{i=1}^{m} \nabla\phi_{I_t^{(i)}}(\mathbf{z}_{k-1})$$
    2. Machine $j$ performs stochastic updates by going through $B_s^{(j)}$ once without replacement:
       $$\mathbf{x}_r \leftarrow \mathbf{x}_{r-1} - \eta\left(\nabla\ell(\mathbf{x}_{r-1}, \xi_l) - \nabla\ell(\mathbf{z}_{k-1}, \xi_l) + \nabla\phi_{I_t}(\mathbf{z}_{k-1}) + \gamma(\mathbf{x}_{r-1} - \mathbf{w}_{t-1})\right)$$
    for $\xi_l \in B_s^{(j)}$.
    3. Machine $j$ update $\mathbf{z}_k$:
       $$\mathbf{z}_k \leftarrow \frac{1}{|B_s^{(j)}|}\sum_{r=0}^{|B_s^{(j)}|} \mathbf{x}_r,$$
    and broadcast $\mathbf{z}_k$ to other machines.
    4. Update indices: $s \leftarrow s + 1$,
    if $s > p_j$ then
      $s \leftarrow 1, \quad j \leftarrow j + 1.$
    end if
  end for
  Update $\mathbf{w}_t \leftarrow \mathbf{z}_K$.
end for
Output: $\mathbf{w}_T$ is the approximate solution.
```

each inner loop, all machines communicate to calculate the global gradient (averaged local gradients) of (12), and then one of the machines j picks a local batch $B_s^{(j)}$ to perform the stochastic updates, where the local batch contains enough samples such that one pass of stochastic updates on $B_s^{(j)}$ decrease the objective quickly. We perform two rounds of communication in each inner loop, one for computing the global gradient, and one for broadcasting the new predictor obtained by a machine j. As we will show in the next section, by carefully choosing the parameters, we will obtain a convergent algorithm for distributed stochastic convex optimization with better efficiency guarantees than previous methods.

We now present detailed analysis for the computation/communication complexity of Algorithm 1 for stochastic quadratic problems, and compare it with related methods in the literature. Throughout this section, we have $\ell(\mathbf{w}, \xi) = \frac{1}{2}(\mathbf{w}^\top \mathbf{x} - y)^2$ where $\xi = (\mathbf{x}, y)$. We assume that

$\ell(\mathbf{w}, \xi)$ is β-smooth and L-Lipschitz in $\mathbf{w}$,[4] and we would like to learn a predictor that is competitive to all predictors with norm at most B. Note that each $\ell(\mathbf{w}, \xi)$ is only weakly convex.

4.1. Efficiency of MP-DSVRG

For the distributed stochastic convex optimization problems, we are concerned with efficiency in terms of sample, communication, computation and memory. Recall that for convex L-Lipshitz, B-bounded problems, to learn a predictor $\hat{\mathbf{w}}$ with ε-generalization error, i.e., $\mathbb{E}\left[\phi(\hat{\mathbf{w}}) - \phi(\mathbf{w}_*)\right] \le \varepsilon$, we require the sample size to be at least $n(\varepsilon) = \mathcal{O}(L^2B^2/\varepsilon^2)$. This sample complexity matches the worst case lower bound, and can be achieved by vanilla SGD.

The theorem below shows that with careful choices of parameters in the outer and inner loops, MP-DSVRG achieves both communication and computation efficiency with the optimal sample complexity.

Theorem 10 (Efficiency of MP-DSVRG) *Set the parameters in Algorithm 1 as follows:*

$$\textit{(outer loop)} \qquad T = \frac{n(\varepsilon)}{bm}, \quad \gamma = \frac{\sqrt{8n(\varepsilon)}L}{bmB}, \quad p_i = \mathcal{O}\left(\frac{\sqrt{n(\varepsilon)}L}{\beta m B}\right)$$

$$\textit{(inner loop)} \qquad K = \mathcal{O}\left(\log n(\varepsilon)\right).$$

Then we have $\mathbb{E}\left[\phi\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t\right) - \phi(\mathbf{w}_*)\right] \le \frac{\sqrt{40}BL}{\sqrt{n(\varepsilon)}} = \mathcal{O}\left(\varepsilon\right).$

Moreover, Algorithm 1 can be implemented with $\mathcal{O}\left(\frac{n(\varepsilon)}{bm}\log n(\varepsilon)\right)$ *rounds of communication, and each machine performs* $\mathcal{O}\left(\frac{n(\varepsilon)}{m}\log n(\varepsilon)\right)$ *vector operations in total.*

We comment on the choice of parameters. For sample efficiency, we fix the sample size $n(\varepsilon)$ and number of machines m, and so we can tradeoff the local minibatch size b and the total number of outer iterations T, maintaining $bT = \frac{n(\varepsilon)}{m}$. For any b, the regularization parameters in the "large minibatch" problem is set to $\gamma = \sqrt{\frac{8T}{bm}} \cdot \frac{L}{B} = \frac{\sqrt{8n(\varepsilon)}L}{bmB}$ according to Theorem 7. Moreover, we choose the number of batches p_i in each local machine in a way that performing one pass of stochastic updates over a single batch by without-replacement sampling is sufficient to reduce the objective by a constant factor.

5. Discussion and conclusion

In this paper, we made progress toward linear speedup, communication and memory efficient methods for distributed stochastic optimization, although we still do not have an algorithm that obtains the "ideal" distributed stochastic optimization performance of linear speedup with constant or near-constant communication and memory. There is also no single known algorithm that dominates all others, with different methods being preferable in terms of different resources. These tradeoffs, up to log-factors, are given in Table 1 and the memory, communication and runtime requirements are also schematically depicted in Figure 2. In the figure, the horizontal axis corresponds to the "minibatch" size, which can be controlled with accelerated minibatch SGD and MP-DSVRG, while other methods are batch methods which consider the entire data set.

4. We can equivalently assume $\|\mathbf{x}\|^2 \le \beta$ and y is bounded.

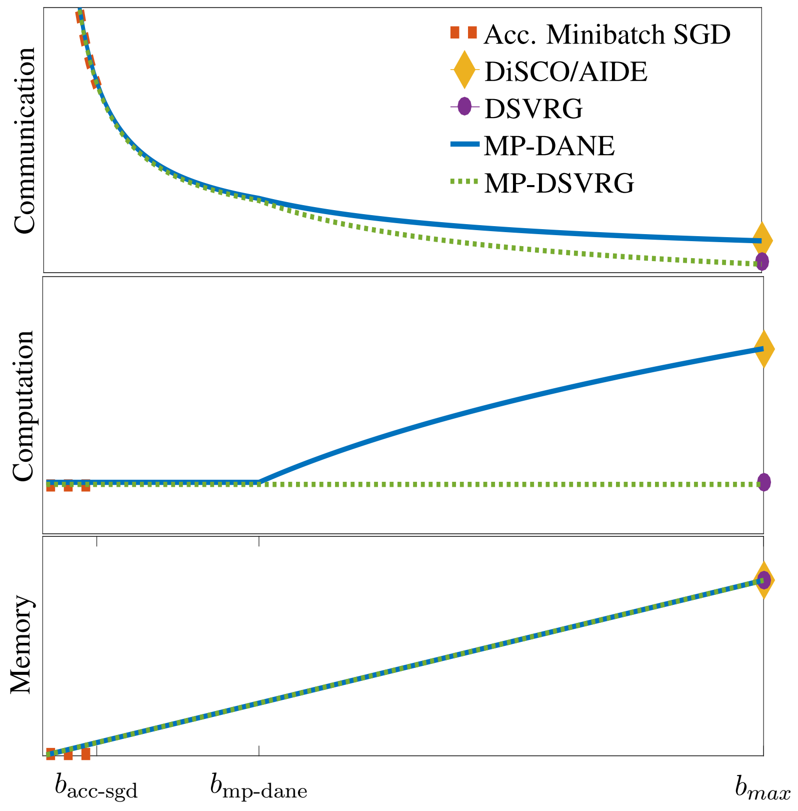

Figure 2: Illustration of theoretical guarantees for MP-DSVRG and the comparison with accelerated minibatch SGD (Cotter et al., 2011), DiSCO (Zhang and Lin, 2015), AIDE (Reddi et al., 2016), DSVRG (Lee et al., 2015), and MP-DANE (proposed and analyzed in Appendix D). We plot the communication, computation and memory requirements while ensuring sample efficiency. Here $b_{\text{acc-sgd}} \asymp n(\varepsilon)^{3/4}/(m\sqrt{B})$, $b_{\text{mp-dane}} \asymp n(\varepsilon)/(m^2B^2)$, and $b_{\max} = n(\varepsilon)/m$.

From Figure 2 we can see that DSVRG (equivalent to MP-DSVRG when $b = n(\varepsilon)/m$) dominates the other methods (up to log-factors) in terms of runtime and communication—it has smaller communication requirements than DiSCO/AIDE (and better than DANE) with nearly the same optimal runtime of accelerated minibatch SGD. But like other batch methods, it requires storing and re-accessing the entire data set. Accelerated minibatch SGD is the only one of these methods requiring only $\mathcal{O}(1)$ memory per machine, and it achieves true linear speedup, but due to the limit on the maximal allowed minibatch size, has relatively high communication cost. MP-DSVRG allows bridging these two extremes of memory and communication, trading off between memory usage and communication. The trade-off is almost an extrapolation, except that in the low-memory high-communication extreme, MP-DSVRG still requires (small) polynomial memory, not minibatch-SGD's $O(1)$ memory, and its runtime still involve a logarithmic factor while minibatch-SGD achieves true linear speedup.

Instead of using DSVRG to solve each proximal subproblem in a minibatch-prox iteration, we can also use any other distributed optimization approach. For example, we can consider using DiSCO or DANE. This is depicted as "MP-DANE" in Figure 2. Again, an external minibatch-prox loop allows trading off memory for communication. For small minibatch sizes, up to a critical value of $b_{\text{mp-dane}} = \Theta(n(\varepsilon)/(m^2 B^2))$, MP-DANE enjoys the same guarantees as MP-DSVRG. But for larger minibatch sizes, such an approach starts suffering from DANE/DiSCO's inferior runtime and communication requirements compared to DSVRG.

We emphasize that the above discussion is based on guarantees established only for least square problems and ignores $\log$-factors. We are unfortunately not aware of distributed stochastic optimization guarantees that improve over minibatch SGD (i.e., achieve even near-linear speedup with lower communication requirements) for general smooth objectives, or achieve true linear speedup (and improved communication guarantees) even for least-square problems.

Acknowledgement

Research was partially supported by an Intel ICRI-CI award and NSF awards IIS 1302662 and BIGDATA 1546500. We would like to thank Ohad Shamir for discussions about Distributed SVRG and Tong Zhang for discussions about minibatch-prox.

References

Dimitri P. Bertsekas. Incremental proximal methods for large scale convex optimization. *Mathematical programming*, 129(2):163, 2011.

Dimitri P. Bertsekas. Incremental aggregated proximal and augmented Lagrangian algorithms. arXiv:1509.09257 [cs.SY], November 4 2015.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Li Cheng, S. V. N. Vishwanathan, Dale Schuurmans, Shaojun Wang, and Terry Caelli. Implicit online learning with kernels. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pages 249–256. MIT Press, 2006.

Andrew Cotter, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *Advances in Neural Information Processing Systems (NIPS)*, volume 24, pages 1647–1655, 2011.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(Mar):551–585, 2006.

Aaron Defazio. A simple practical accelerated method for finite sums. In *Advances In Neural Information Processing Systems*, pages 676–684, 2016.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(Jan):165–202, 2012.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

Brian Kulis and Peter L. Bartlett. Implicit online learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 575–582, 2010.

Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an $o(1/t)$ convergence rate for the projected stochastic subgradient method. arXiv:1212.2002 [cs.LG], 2012.

Jason D Lee, Qihang Lin, Tengyu Ma, and Tianbao Yang. Distributed stochastic variance reduced gradient methods and a lower bound for communication complexity. *arXiv preprint arXiv:1507.07595*, 2015.

Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J. Smola. Efficient mini-batch training for stochastic optimization. In *Proc. of the 20th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (SIGKDD 2014)*, pages 661–670, 2014.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3384–3392, 2015.

Deanna Needell and Joel A. Tropp. Paved with good intentions: Analysis of a randomized block kaczmarz method. *Linear Algebra and its Applications*, 441:199–221, 2014.

A. Nemirovskii and D. B. Yudin. Problem complexity and method efficiency in optimization, 1983.

Sashank J Reddi, Jakub Konečnỳ, Peter Richtárik, Barnabás Póczós, and Alex Smola. Aide: Fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*, 2016.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 24, pages 1458–1466, 2011.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *Proc. of the 22th Annual Conference on Learning Theory (COLT'09)*, 2009.

Ohad Shamir. Without-replacement sampling for stochastic gradient methods: Convergence results and application to distributed optimization. *arXiv preprint arXiv:1603.00570*, 2016.

Ohad Shamir and Nathan Srebro. Distributed stochastic optimization and learning. In *52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 850–857. IEEE, 2014.

Ohad Shamir, Nathan Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate Newton-type method. In *Proc. of the 31st Int. Conf. Machine Learning (ICML 2014)*, pages 1000–1008, 2014.

Panos Toulis and Edoardo M. Airoldi. Implicit stochastic gradient descent. *arXiv preprint arXiv:1408.2923*, 2014.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1995.

Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Yuchen Zhang and Xiao Lin. DiSCO: Distributed optimization for self-concordant empirical loss. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 362–370, 2015.

Yuchen Zhang, Martin J. Wainwright, and John C. Duchi. Communication-efficient algorithms for statistical optimization. In *Advances in Neural Information Processing Systems*, pages 1502–1510, 2012.

Appendix A. Analysis of exact minibatch-prox

A.1. Proof of Lemma 1

Proof Observe that (4) implies $\gamma_t(\mathbf{w}_{t-1} - \mathbf{w}_t)$ is a subgradient at $\mathbf{w}_t$ of the sum of $\phi_{I_t}(\mathbf{w})$ and the indicator function of Ω (which has value 0 in Ω and ∞ otherwise), and thus we have for any $\mathbf{w} \in \Omega$ that

$$\phi_{I_t}(\mathbf{w}) - \phi_{I_t}(\mathbf{w}_t) \geq \gamma_t \left\langle \mathbf{w}_{t-1} - \mathbf{w}_t,\, \mathbf{w} - \mathbf{w}_t \right\rangle + \frac{\lambda}{2} \left\| \mathbf{w} - \mathbf{w}_t \right\|^2 . \tag{14}$$

For any $\mathbf{w} \in \Omega$, we can bound its distance to $\mathbf{w}_{t-1}$ as

$$\begin{aligned}
\left\| \mathbf{w}_{t-1} - \mathbf{w} \right\|^2 &= \left\| \mathbf{w}_{t-1} - \mathbf{w}_t + \mathbf{w}_t - \mathbf{w} \right\|^2 \\
&= \left\| \mathbf{w}_{t-1} - \mathbf{w}_t \right\|^2 + 2 \left\langle \mathbf{w}_{t-1} - \mathbf{w}_t,\, \mathbf{w}_t - \mathbf{w} \right\rangle + \left\| \mathbf{w}_t - \mathbf{w} \right\|^2 \\
&\geq \left\| \mathbf{w}_{t-1} - \mathbf{w}_t \right\|^2 + \frac{2}{\gamma_t} \left(\phi_{I_t}(\mathbf{w}_t) - \phi_{I_t}(\mathbf{w}) \right) + \frac{\lambda}{\gamma_t} \left\| \mathbf{w} - \mathbf{w}_t \right\|^2 + \left\| \mathbf{w}_t - \mathbf{w} \right\|^2 \\
&= \frac{\lambda + \gamma_t}{\gamma_t} \left\| \mathbf{w}_t - \mathbf{w} \right\|^2 + \frac{2}{\gamma_t} \left(\phi_{I_t}(\mathbf{w}_t) - \phi_{I_t}(\mathbf{w}) \right) + \left\| \mathbf{w}_{t-1} - \mathbf{w}_t \right\|^2
\end{aligned}$$

where we have used (14) in the first inequality. Rearranging the terms yields the desired result. ■

A.2. Proof of Lemma 2

The following lemma, which is essentially shown by Shalev-Shwartz et al. (2009, Theorem 6), characterizes the convergence of the empirical loss to the population counterpart for the (approximate) regularized empirical risk minimizer.

Lemma 11 *Let the instantaneous function $\ell(\mathbf{w}, \xi)$ be L-Lipschitz and λ-strongly convex in $\mathbf{w}$. Consider the following regularized ERM problem with sample set $Z = \{\xi_1, \dots, \xi_n\}$:*

$$\hat{\mathbf{w}} = \operatorname*{arg\,min}_{\mathbf{w} \in \Omega} \ \hat{F}(\mathbf{w}) \qquad \textit{where} \quad \hat{F}(\mathbf{w}) := \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \xi_i) + r(\mathbf{w}),$$

and the regularizer $r(\mathbf{w})$ is γ-strongly convex. Denote by $G(\mathbf{w}) = \mathbb{E}_{\xi}\left[\ell(\mathbf{w}, \xi)\right]$ and $\hat{G}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}, \xi_i)$ the expected and the empirical losses respectively.

1. *For the regularized empirical risk minimizer $\hat{\mathbf{w}}$, we have*

$$\left| \mathbb{E}_Z \left[G(\hat{\mathbf{w}}) - \hat{G}(\hat{\mathbf{w}}) \right] \right| \le \frac{4L^2}{(\lambda + \gamma) n}.$$

2. *If for any given dataset Z, a possibly randomized algorithm $\mathcal{A}$ minimizes $\hat{F}(\mathbf{w})$ up to an error of η, i.e., $\mathcal{A}$ returns an approximate solution $\tilde{\mathbf{w}}$ such that $\mathbb{E}_{\mathcal{A}}\left[\hat{F}(\tilde{\mathbf{w}}) - \hat{F}(\hat{\mathbf{w}})\right] \le \eta$, we have*

$$\left| \mathbb{E}_{Z,\mathcal{A}} \left[G(\tilde{\mathbf{w}}) - \hat{G}(\hat{\mathbf{w}}) \right] \right| \le \frac{4L^2}{(\lambda + \gamma) n} + \sqrt{\frac{2L^2 \eta}{\lambda + \gamma}}.$$

Proof We prove the lemma by a stability argument.

Exact ERM Denote by $Z^{(i)}$ the sample set that is identical to Z except that the i-th sample ξ_i is replaced by another random sample ξ_i', by $\hat{F}^{(i)}(\mathbf{w})$ the empirical objective defined using $Z^{(i)}$, i.e.,

$$\hat{F}^{(i)}(\mathbf{w}) := \frac{1}{n} \left(\sum_{j \ne i} \ell(\mathbf{w}, \xi_i) + \ell(\mathbf{w}, \xi_i') \right) + r(\mathbf{w}),$$

and by $\hat{\mathbf{w}}^{(i)} = \operatorname{arg\,min}_{\mathbf{w} \in \Omega} \ \hat{F}^{(i)}(\mathbf{w})$ the empirical risk minimizer of $\hat{F}^{(i)}(\mathbf{w})$.

By the definition of the empirical objectives, we have

$$\begin{aligned}
\hat{F}(\hat{\mathbf{w}}^{(i)}) - \hat{F}(\hat{\mathbf{w}}) &= \frac{\ell(\hat{\mathbf{w}}^{(i)}, \xi_i) - \ell(\hat{\mathbf{w}}, \xi_i)}{n} + \frac{\sum_{j \ne i} \ell(\hat{\mathbf{w}}^{(i)}, \xi_i) - \ell(\hat{\mathbf{w}}, \xi_i)}{n} + r(\hat{\mathbf{w}}^{(i)}) - r(\hat{\mathbf{w}}) \\
&= \frac{\ell(\hat{\mathbf{w}}^{(i)}, \xi_i) - \ell(\hat{\mathbf{w}}, \xi_i)}{n} + \frac{\ell(\hat{\mathbf{w}}, \xi_i') - \ell(\hat{\mathbf{w}}^{(i)}, \xi_i')}{n} + \left(\hat{F}^{(i)}(\hat{\mathbf{w}}^{(i)}) - \hat{F}^{(i)}(\hat{\mathbf{w}}) \right) \\
&\le \frac{\left| \ell(\hat{\mathbf{w}}^{(i)}, \xi_i) - \ell(\hat{\mathbf{w}}, \xi_i) \right|}{n} + \frac{\left| \ell(\hat{\mathbf{w}}, \xi_i') - \ell(\hat{\mathbf{w}}^{(i)}, \xi_i') \right|}{n} \\
&\le \frac{2L}{n} \left\| \hat{\mathbf{w}}^{(i)} - \hat{\mathbf{w}} \right\| \qquad (15)
\end{aligned}$$

where we have used the fact that $\hat{\mathbf{w}}^{(i)}$ is the minimizer of $\hat{F}^{(i)}(\mathbf{w})$ in the first inequality, and the L-Lipschitz continuity of $\ell(\mathbf{w}, \xi)$ in the second inequality.

On the other hand, it follows from the $(\lambda + \gamma)$-strong convexity of $\hat{F}(\mathbf{w})$ that

$$\hat{F}(\hat{\mathbf{w}}^{(i)}) - \hat{F}(\hat{\mathbf{w}}) \ge \frac{(\lambda + \gamma)}{2} \left\| \hat{\mathbf{w}}^{(i)} - \hat{\mathbf{w}} \right\|^2. \qquad (16)$$

Combining (15) and (16) yields $\left\| \hat{\mathbf{w}}^{(i)} - \hat{\mathbf{w}} \right\| \le \frac{4L}{(\lambda+\gamma)n}$.

Again, by the L-Lipschitz continuity of $\ell(\mathbf{w}, \xi)$, we have that for any sample ξ that

$$\left| \ell(\hat{\mathbf{w}}, \xi) - \ell(\hat{\mathbf{w}}^{(i)}, \xi) \right| \le L \left\| \hat{\mathbf{w}}^{(i)} - \hat{\mathbf{w}} \right\| \le \frac{4L^2}{(\lambda+\gamma)n}. \tag{17}$$

Since Z and $Z^{(i)}$ are both i.i.d. sample sets, we have

$$\mathbb{E}_Z \left[G(\hat{\mathbf{w}}) \right] = \mathbb{E}_{Z^{(i)}} \left[G(\hat{\mathbf{w}}^{(i)}) \right] = \mathbb{E}_{Z^{(i)} \cup \{\xi_i\}} \left[\ell(\hat{\mathbf{w}}^{(i)}, \xi_i) \right].$$

As this holds for all $i = 1, \dots, n$, we can also write

$$\mathbb{E}_Z \left[G(\hat{\mathbf{w}}) \right] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{Z^{(i)} \cup \{\xi_i\}} \left[\ell(\hat{\mathbf{w}}^{(i)}, \xi_i) \right]. \tag{18}$$

On the other hand, we have

$$\mathbb{E}_Z \left[\hat{G}(\hat{\mathbf{w}}) \right] = \mathbb{E}_Z \left[\frac{1}{n} \sum_{i=1}^{n} \ell(\hat{\mathbf{w}}, \xi_i) \right] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_Z \left[\ell(\hat{\mathbf{w}}, \xi_i) \right]. \tag{19}$$

Combining (18) and (19) and using the stability (17), we obtain

$$\mathbb{E}_Z \left[G(\hat{\mathbf{w}}) - \hat{G}(\hat{\mathbf{w}}) \right] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{Z \cup \{\xi_i'\}} \left[\ell(\hat{\mathbf{w}}^{(i)}, \xi_i) - \ell(\hat{\mathbf{w}}, \xi_i) \right] \in \left[-\frac{4L^2}{(\lambda+\gamma)n}, \frac{4L^2}{(\lambda+\gamma)n} \right].$$

Inexact ERM For the approximate solution $\tilde{\mathbf{w}}$, due to the $(\lambda+\gamma)$-strong convexity of $\hat{F}(\mathbf{w})$, we have

$$\mathbb{E}_{\mathcal{A}} \left\| \tilde{\mathbf{w}} - \hat{\mathbf{w}} \right\|^2 \le \frac{2}{\lambda+\gamma} \mathbb{E}_{\mathcal{A}} \left[\hat{F}(\tilde{\mathbf{w}}) - \hat{F}(\hat{\mathbf{w}}) \right] \le \frac{2\eta}{\lambda+\gamma},$$

and thus $\mathbb{E}_{\mathcal{A}} \left\| \tilde{\mathbf{w}} - \hat{\mathbf{w}} \right\| \le \sqrt{\frac{2\eta}{\lambda+\gamma}}$ by the fact that $\mathbb{E}x^2 \ge (\mathbb{E}x)^2$ for any random variable x.

It then follows from the Lipschitz continuity of $G(\mathbf{w})$ that

$$\mathbb{E}_{\mathcal{A}} \left| G(\tilde{\mathbf{w}}) - G(\hat{\mathbf{w}}) \right| \le L \cdot \mathbb{E}_{\mathcal{A}} \left\| \tilde{\mathbf{w}} - \hat{\mathbf{w}} \right\| \le \sqrt{\frac{2L^2\eta}{\lambda+\gamma}}.$$

Finally, we have by the triangle inequality and the stability of exact ERM that

$$\begin{aligned} \left| \mathbb{E}_{Z,\mathcal{A}} \left[G(\tilde{\mathbf{w}}) - \hat{G}(\hat{\mathbf{w}}) \right] \right| &\le \mathbb{E}_Z \left[\mathbb{E}_{\mathcal{A}} \left| G(\tilde{\mathbf{w}}) - G(\hat{\mathbf{w}}) \right| \right] + \left| \mathbb{E}_Z \left[G(\hat{\mathbf{w}}) - \hat{G}(\hat{\mathbf{w}}) \right] \right| \\ &\le \sqrt{\frac{2L^2\eta}{\lambda+\gamma}} + \frac{4L^2}{(\lambda+\gamma)n}. \end{aligned}$$

■

Then Lemma 2 follows from the fact that that our stochastic objective (8) is equipped with L-Lipschitz, λ-strongly convex loss $\phi(\mathbf{w})$ and γ_t-strongly convex regularizer $\frac{\gamma_t}{2} \left\| \mathbf{w} - \mathbf{w}_{t-1} \right\|^2$.

A.3. Proof of Lemma 3

Proof We have by Lemma 2 that

$$\left|\mathbb{E}_{I_t}\left[\phi_{I_t}(\mathbf{w}_t) - \phi(\mathbf{w}_t)\right]\right| \le \frac{4L^2}{(\lambda+\gamma_t)b}.$$

Take expectation of (6) over the random sampling of I_t and we obtain

$$\begin{aligned}
\frac{\lambda+\gamma_t}{\gamma_t}\mathbb{E}_{I_t}\left\|\mathbf{w}_t - \mathbf{w}\right\|^2 &\le \left\|\mathbf{w}_{t-1} - \mathbf{w}\right\|^2 - \frac{2}{\gamma_t}\left(\mathbb{E}_{I_t}\left[\phi_{I_t}(\mathbf{w}_t)\right] - \phi(\mathbf{w})\right) \\
&= \left\|\mathbf{w}_{t-1} - \mathbf{w}\right\|^2 - \frac{2}{\gamma_t}\left(\mathbb{E}_{I_t}\left[\phi_{I_t}(\mathbf{w}_t) - \phi(\mathbf{w}_t)\right] + \mathbb{E}_{I_t}\left[\phi(\mathbf{w}_t) - \phi(\mathbf{w})\right]\right) \\
&\le \left\|\mathbf{w}_{t-1} - \mathbf{w}\right\|^2 - \frac{2}{\gamma_t}\mathbb{E}_{I_t}\left[\phi(\mathbf{w}_t) - \phi(\mathbf{w})\right] + \frac{2}{\gamma_t}\left|\mathbb{E}_{I_t}\left[\phi_{I_t}(\mathbf{w}_t) - \phi(\mathbf{w}_t)\right]\right| \\
&\le \left\|\mathbf{w}_{t-1} - \mathbf{w}\right\|^2 - \frac{2}{\gamma_t}\mathbb{E}_{I_t}\left[\phi(\mathbf{w}_t) - \phi(\mathbf{w})\right] + \frac{8L^2}{\gamma_t(\lambda+\gamma_t)b}.
\end{aligned}$$

■

A.4. Proof of Theorem 4

Proof When $\ell(\mathbf{w}, \xi)$ is weakly convex (i.e., $\lambda = 0$), we further set $\gamma_t = \gamma$ for all $t \ge 1$. Applying Lemma 3 with $\mathbf{w} = \mathbf{w}_*$ yields

$$\mathbb{E}_{I_t}\left[\phi(\mathbf{w}_t) - \phi(\mathbf{w}_*)\right] \le \frac{\gamma}{2}\left(\left\|\mathbf{w}_{t-1} - \mathbf{w}_*\right\|^2 - \mathbb{E}_{I_t}\left\|\mathbf{w}_t - \mathbf{w}_*\right\|^2\right) + \frac{4L^2}{\gamma b}. \tag{20}$$

Summing (20) for $t = 1, \dots, T$ yields

$$\sum_{t=1}^{T}\mathbb{E}\left[\phi(\mathbf{w}_t) - \phi(\mathbf{w}_*)\right] \le \frac{\gamma}{2}\left\|\mathbf{w}_0 - \mathbf{w}_*\right\|^2 + \frac{4L^2T}{\gamma b}.$$

Minimizing the RHS over γ gives the optimal choice

$$\gamma = \sqrt{\frac{8T}{b}} \cdot \frac{L}{\left\|\mathbf{w}_0 - \mathbf{w}_*\right\|},$$

with a corresponding regret

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\phi(\mathbf{w}_t) - \phi(\mathbf{w}_*)\right] \le \frac{\sqrt{8}L}{\sqrt{bT}}\left\|\mathbf{w}_0 - \mathbf{w}_*\right\|.$$

As a result, by returning the uniform average $\widehat{\mathbf{w}}_T = \frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t$, we have due to the convexity of $\phi(\mathbf{w})$ that

$$\mathbb{E}\left[\phi(\widehat{\mathbf{w}}_T) - \phi(\mathbf{w}_*)\right] \le \frac{\sqrt{8}L}{\sqrt{bT}}\left\|\mathbf{w}_0 - \mathbf{w}_*\right\|.$$

■

A.5. Proof of Theorem 5

Proof Let $\ell(\mathbf{w}, \xi)$ be λ-strongly convex for some $\lambda > 0$. Applying Lemma 3 with $\mathbf{w} = \mathbf{w}_*$ yields

$$\mathbb{E}_{I_t}\left[\phi(\mathbf{w}_t) - \phi(\mathbf{w}_*)\right] \le \left(\frac{\gamma_t}{2}\left\|\mathbf{w}_{t-1} - \mathbf{w}_*\right\|^2 - \frac{\lambda + \gamma_t}{2}\mathbb{E}_{I_t}\left\|\mathbf{w}_t - \mathbf{w}_*\right\|^2\right) + \frac{4L^2}{(\lambda + \gamma_t)b}. \tag{21}$$

Setting $\gamma_t = \frac{\lambda(t-1)}{2}$ for $t = 1, \dots,$[5], the above inequality becomes

$$\begin{aligned}\mathbb{E}_{I_t}\left[\phi(\mathbf{w}_t) - \phi(\mathbf{w}_*)\right] &\le \left(\frac{\lambda(t-1)}{4}\left\|\mathbf{w}_{t-1} - \mathbf{w}_*\right\|^2 - \frac{\lambda(t+1)}{4}\mathbb{E}_{I_t}\left\|\mathbf{w}_t - \mathbf{w}_*\right\|^2\right) + \frac{8L^2}{\lambda b(t+1)} \\ &\le \left(\frac{\lambda(t-1)}{4}\left\|\mathbf{w}_{t-1} - \mathbf{w}_*\right\|^2 - \frac{\lambda(t+1)}{4}\mathbb{E}_{I_t}\left\|\mathbf{w}_t - \mathbf{w}_*\right\|^2\right) + \frac{8L^2}{\lambda bt},\end{aligned}$$

and therefore

$$t \cdot \mathbb{E}_{I_t}\left[\phi(\mathbf{w}_t) - \phi(\mathbf{w}_*)\right] \le \frac{\lambda}{4}\left((t-1)t\left\|\mathbf{w}_{t-1} - \mathbf{w}_*\right\|^2 - t(t+1)\mathbb{E}_{I_t}\left\|\mathbf{w}_t - \mathbf{w}_*\right\|^2\right) + \frac{8L^2}{\lambda b}.$$

Summing this inequality for $t = 1, \dots, T$ yields

$$\sum_{t=1}^{T} t \cdot \mathbb{E}\left[\phi(\mathbf{w}_t) - \phi(\mathbf{w}_*)\right] \le \frac{8L^2T}{\lambda b}.$$

As a result, by returning the weighted average $\widehat{\mathbf{w}}_T = \frac{2}{T(T+1)}\sum_{t=1}^{T} t\mathbf{w}_t$, we have due to the convexity of $\phi(\mathbf{w})$ that $\phi(\widehat{\mathbf{w}}_T) \le \frac{2}{T(T+1)}\sum_{t=1}^{T} t \cdot \phi(\mathbf{w}_t)$ and

$$\mathbb{E}\left[\phi(\widehat{\mathbf{w}}_T) - \phi(\mathbf{w}_*)\right] \le \frac{2}{T(T+1)}\sum_{t=1}^{T} t \cdot \mathbb{E}\left[\phi(\mathbf{w}_t) - \phi(\mathbf{w}_*)\right] \le \frac{16L^2}{\lambda b(T+1)}.$$

■

Appendix B. Analysis of inexact minibatch-prox

B.1. Proof of Lemma 6

Proof Due to the $(\lambda + \gamma_t)$-strong convexity of $\tilde{f}_t(\mathbf{w})$, we have

$$\mathbb{E}_{\mathcal{A}}\left\|\tilde{\mathbf{w}}_t - \bar{\mathbf{w}}_t\right\|^2 \le \frac{2}{\lambda + \gamma_t}\mathbb{E}_{\mathcal{A}}\left[\tilde{f}_t(\tilde{\mathbf{w}}_t) - \tilde{f}_t(\bar{\mathbf{w}}_t)\right] \le \frac{2\eta_t}{\lambda + \gamma_t}.$$

Applying Lemma 1 to the exact minimizer $\bar{\mathbf{w}}_t$ yields

$$\phi_{I_t}(\bar{\mathbf{w}}_t) - \phi_{I_t}(\mathbf{w}) \le \frac{\gamma_t}{2}\left\|\tilde{\mathbf{w}}_{t-1} - \mathbf{w}\right\|^2 - \frac{\lambda + \gamma_t}{2}\left\|\bar{\mathbf{w}}_t - \mathbf{w}\right\|^2.$$

5. This choice is inspired by the stepsize rule of Lacoste-Julien et al. (2012) for stochastic gradient descent.

Therefore, for the t-th iteration, we have

$$
\begin{aligned}
& \mathbb{E}_{I_t,\mathcal{A}} \left[\phi(\tilde{\mathbf{w}}_t) - \phi(\mathbf{w}) \right] \\
= & \mathbb{E}_{I_t,\mathcal{A}} \left[\phi(\tilde{\mathbf{w}}_t) - \phi_{I_t}(\bar{\mathbf{w}}_t) \right] + \mathbb{E}_{I_t} \left[\phi_{I_t}(\bar{\mathbf{w}}_t) - \phi_{I_t}(\mathbf{w}) \right] \\
\le & \frac{4L^2}{(\lambda+\gamma_t)b} + \sqrt{\frac{2L^2\eta_t}{\lambda+\gamma_t}} + \frac{\gamma_t}{2} \left\| \tilde{\mathbf{w}}_{t-1} - \mathbf{w} \right\|^2 - \frac{\lambda+\gamma_t}{2} \mathbb{E}_{I_t} \left\| \bar{\mathbf{w}}_t - \mathbf{w} \right\|^2 \\
\le & \frac{4L^2}{(\lambda+\gamma_t)b} + \sqrt{\frac{2L^2\eta_t}{\lambda+\gamma_t}} + \frac{\gamma_t}{2} \left\| \tilde{\mathbf{w}}_{t-1} - \mathbf{w} \right\|^2 - \frac{\lambda+\gamma_t}{2} \mathbb{E}_{I_t,\mathcal{A}} \left(\left\| \tilde{\mathbf{w}}_t - \mathbf{w} \right\| - \left\| \tilde{\mathbf{w}}_t - \bar{\mathbf{w}}_t \right\| \right)^2 \\
\le & \frac{4L^2}{(\lambda+\gamma_t)b} + \sqrt{\frac{2L^2\eta_t}{\lambda+\gamma_t}} + \frac{\gamma_t}{2} \left\| \tilde{\mathbf{w}}_{t-1} - \mathbf{w} \right\|^2 - \frac{\lambda+\gamma_t}{2} \mathbb{E}_{I_t,\mathcal{A}} \left\| \tilde{\mathbf{w}}_t - \mathbf{w} \right\|^2 \\
& \qquad + (\lambda+\gamma_t) \cdot \mathbb{E}_{I_t,\mathcal{A}} \left[\left\| \tilde{\mathbf{w}}_t - \bar{\mathbf{w}}_t \right\| \cdot \left\| \tilde{\mathbf{w}}_t - \mathbf{w} \right\| \right] \\
\le & \frac{4L^2}{(\lambda+\gamma_t)b} + \sqrt{\frac{2L^2\eta_t}{\lambda+\gamma_t}} + \frac{\gamma_t}{2} \left\| \tilde{\mathbf{w}}_{t-1} - \mathbf{w} \right\|^2 - \frac{\lambda+\gamma_t}{2} \mathbb{E}_{I_t,\mathcal{A}} \left\| \tilde{\mathbf{w}}_t - \mathbf{w} \right\|^2 \\
& \qquad + (\lambda+\gamma_t) \sqrt{\mathbb{E}_{I_t,\mathcal{A}} \left\| \tilde{\mathbf{w}}_t - \bar{\mathbf{w}}_t \right\|^2} \cdot \sqrt{\mathbb{E}_{I_t,\mathcal{A}} \left\| \tilde{\mathbf{w}}_t - \mathbf{w} \right\|^2} \\
\le & \frac{4L^2}{(\lambda+\gamma_t)b} + \sqrt{\frac{2L^2\eta_t}{\lambda+\gamma_t}} + \frac{\gamma_t}{2} \left\| \tilde{\mathbf{w}}_{t-1} - \mathbf{w} \right\|^2 - \frac{\lambda+\gamma_t}{2} \mathbb{E}_{I_t,\mathcal{A}} \left\| \tilde{\mathbf{w}}_t - \mathbf{w} \right\|^2 \\
& \qquad + \sqrt{2(\lambda+\gamma_t)\eta_t} \cdot \sqrt{\mathbb{E}_{I_t,\mathcal{A}} \left\| \tilde{\mathbf{w}}_t - \mathbf{w} \right\|^2}
\end{aligned}
$$

where we have applied Lemma 11 to the approximate minimizer $\tilde{\mathbf{w}}_t$ in the first inequality, used the triangle inequality $\|\bar{\mathbf{w}}_t - \mathbf{w}\| \ge \left| \|\tilde{\mathbf{w}}_t - \mathbf{w}\| - \|\tilde{\mathbf{w}}_t - \bar{\mathbf{w}}_t\| \right|$ in the second inequality, dropped a negative term in the third inequality, and used the Cauchy-Schwarz inequality for random variables in the fourth inequality. ■

B.2. Proof of Theorem 7

When $\ell(\mathbf{w}, \xi)$ is weakly convex (i.e., $\lambda = 0$), set $\gamma_t = \gamma$ for all $t \ge 1$ as in exact minibatch-prox. Then summing (11) for $t = 1, \dots, T$ yields

$$
\begin{aligned}
\sum_{t=1}^{T} \mathbb{E} \left[\phi(\tilde{\mathbf{w}}_t) - \phi(\mathbf{w}_*) \right] + \frac{\gamma}{2} \mathbb{E} \left\| \tilde{\mathbf{w}}_T - \mathbf{w}_* \right\|^2 \le \frac{\gamma}{2} \left\| \tilde{\mathbf{w}}_0 - \mathbf{w}_* \right\|^2 + \frac{4L^2 T}{\gamma b} + \sum_{t=1}^{T} \sqrt{\frac{2L^2\eta_t}{\gamma}} \\
+ \sum_{t=1}^{T} \sqrt{2\gamma\eta_t} \cdot \sqrt{\mathbb{E} \left\| \tilde{\mathbf{w}}_t - \mathbf{w}_* \right\|^2} \qquad (22)
\end{aligned}
$$

where the expectation is taken over random sampling and the randomness of $\mathcal{A}$ in the first T iterations. To resolve the recursion, we need the following lemma by Schmidt et al. (2011).

Lemma 12 *Assume that the non-negative sequence $\{u_T\}$ satisfies the following recursion for all $T \geq 1$:*

$$u_T^2 \leq S_T + \sum_{t=1}^{T} \lambda_t u_t,$$

with S_T an increasing sequence, $S_0 \geq u_0^2$ and $\lambda_t \geq 0$ for all t. Then, for all $T \geq 1$, we have

$$u_T \leq \frac{1}{2}\sum_{t=1}^{T} \lambda_t + \left(S_T + \left(\frac{1}{2}\sum_{t=1}^{T} \lambda_t \right)^2 \right)^{\frac{1}{2}} \leq \sqrt{S_T} + \sum_{t=1}^{T} \lambda_t.$$

We are now ready to prove Theorem 7.

Proof Bounding $\sqrt{\mathbb{E}\left\|\tilde{\mathbf{w}}_t - \mathbf{w}_*\right\|^2}$. Dropping the $\sum_{t=1}^{T} \mathbb{E}\left[\phi(\tilde{\mathbf{w}}_t) - \phi(\mathbf{w}_*)\right]$ term from (22) which is non-negative due to the optimality of $\mathbf{w}_*$, we obtain

$$\mathbb{E}\left\|\tilde{\mathbf{w}}_T - \mathbf{w}_*\right\|^2 \leq \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|^2 + \frac{8L^2T}{\gamma^2 b} + \sum_{t=1}^{T} \sqrt{\frac{8L^2\eta_t}{\gamma^3}} + \sum_{t=1}^{T} \sqrt{\frac{8\eta_t}{\gamma}} \cdot \sqrt{\mathbb{E}\left\|\tilde{\mathbf{w}}_t - \mathbf{w}_*\right\|^2}.$$

Now apply Lemma 12 (using $u_T = \sqrt{\mathbb{E}\left\|\tilde{\mathbf{w}}_T - \mathbf{w}_*\right\|^2}$, $S_T = \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|^2 + \frac{8L^2T}{\gamma^2 b} + \sum_{t=1}^{T} \sqrt{\frac{8L^2\eta_t}{\gamma^3}}$, and $\lambda_t = \sqrt{\frac{8\eta_t}{\gamma}}$) and the fact that $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$ for $x,\ y \geq 0$, we have

$$\sqrt{\mathbb{E}\left\|\tilde{\mathbf{w}}_T - \mathbf{w}_*\right\|^2} \leq \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\| + \sqrt{\frac{8L^2T}{\gamma^2 b}} + \sum_{t=1}^{T} \sqrt{\frac{8\eta_t}{\gamma}} + \sqrt{\sum_{t=1}^{T} \sqrt{\frac{8L^2\eta_t}{\gamma^3}}}$$

We have thus bounded the sequence of $\sqrt{\mathbb{E}\left\|\tilde{\mathbf{w}}_T - \mathbf{w}_*\right\|^2}$ by a non-negative increasing sequence.

Bounding function values. Dropping the $\mathbb{E}\left\|\tilde{\mathbf{w}}_T - \mathbf{w}_*\right\|^2$ term from (22) which is non-negative, we obtain

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{E}\left[\phi(\tilde{\mathbf{w}}_t) - \phi(\mathbf{w}_*)\right] \\
&\leq \frac{\gamma}{2}\left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|^2 + \frac{4L^2T}{\gamma b} + \sum_{t=1}^{T} \sqrt{\frac{2L^2\eta_t}{\gamma}} + \sum_{t=1}^{T} \sqrt{2\eta_t\gamma} \cdot \sqrt{\mathbb{E}\left\|\tilde{\mathbf{w}}_t - \mathbf{w}_*\right\|^2} \\
&\leq \frac{\gamma}{2}\left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|^2 + \frac{4L^2T}{\gamma b} + \sum_{t=1}^{T} \sqrt{\frac{2L^2\eta_t}{\gamma}} + \left(\sum_{t=1}^{T} \sqrt{2\eta_t\gamma}\right) \cdot \max_{1\leq t\leq T} \sqrt{\mathbb{E}\left\|\tilde{\mathbf{w}}_t - \mathbf{w}_*\right\|^2} \\
&\leq \frac{\gamma}{2}\left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|^2 + \frac{4L^2T}{\gamma b} + \sum_{t=1}^{T} \sqrt{\frac{2L^2\eta_t}{\gamma}} \\
&\qquad + \left(\sum_{t=1}^{T} \sqrt{2\eta_t\gamma}\right) \cdot \left(\left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\| + \sqrt{\frac{8L^2T}{\gamma^2 b}} + \sum_{t=1}^{T} \sqrt{\frac{8\eta_t}{\gamma}} + \sqrt{\sum_{t=1}^{T} \sqrt{\frac{8L^2\eta_t}{\gamma^3}}} \right). \qquad (23)
\end{aligned}$$

To achieve the same order of regret as in exact minibatch-prox, we require that η_t decays with t, and in particular

$$\eta_t \le \min\left(c_1 \left(\frac{T}{b}\right)^{\frac{1}{2}},\, c_2 \left(\frac{T}{b}\right)^{\frac{3}{2}}\right) \cdot \frac{L \left\| \tilde{\mathbf{w}}_0 - \mathbf{w}_* \right\|}{t^{2+2\delta}} \tag{24}$$

for some $\delta > 0$. Note that η_t has the unit of function value. Let $c := \sum_{i=1}^{\infty} \frac{1}{i^{1+\delta}} \le \frac{1+\delta}{\delta}$ which only depends on δ (as a concrete example, we have $c = \frac{\pi^2}{6}$ when $\delta = 2$).

Using the choice of $\gamma = \sqrt{\frac{8T}{b}} \cdot \frac{L}{\left\|\mathbf{w}_0 - \mathbf{w}_*\right\|}$, we obtain from (24) that

$$\begin{aligned}
\sum_{t=1}^{T} \sqrt{\frac{8\eta_t}{\gamma}} &= \sum_{t=1}^{T} \sqrt{\sqrt{\frac{8b}{T}} \cdot \frac{\left\|\mathbf{w}_0 - \mathbf{w}_*\right\|}{L} \cdot \eta_t} \le 8^{\frac{1}{4}} c_1^{\frac{1}{2}} \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\| \sum_{t=1}^{T} \frac{1}{t^{1+\delta}} \\
&\le 8^{\frac{1}{4}} c_1^{\frac{1}{2}} c \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|, \\
\sum_{t=1}^{T} \sqrt{\frac{8L^2\eta_t}{\gamma^3}} &= \sum_{t=1}^{T} \sqrt{\sqrt{\frac{b^3}{8T^3}} \cdot \frac{\left\|\mathbf{w}_0 - \mathbf{w}_*\right\|^3}{L} \cdot \eta_t} \le 8^{-\frac{1}{4}} c_2^{\frac{1}{2}} \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|^2 \sum_{t=1}^{T} \frac{1}{t^{1+\delta}} \\
&\le 8^{-\frac{1}{4}} c_2^{\frac{1}{2}} c \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|^2.
\end{aligned}$$

Continuing from (23) and substituting in the value of γ, we have

$$\begin{aligned}
\sum_{t=1}^{T} \mathbb{E}\left[\phi(\tilde{\mathbf{w}}_t) - \phi(\mathbf{w}_*)\right] &\le \sqrt{\frac{8T}{b}} \cdot L \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\| + \frac{\gamma}{2} \sum_{t=1}^{T} \sqrt{\frac{8L^2\eta_t}{\gamma^3}} \\
&\quad + \frac{\gamma}{2} \left(\sum_{t=1}^{T} \sqrt{\frac{8\eta_t}{\gamma}}\right) \cdot \left(2\left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\| + \sum_{t=1}^{T} \sqrt{\frac{8\eta_t}{\gamma}} + \sqrt{\sum_{t=1}^{T} \sqrt{\frac{8L^2\eta_t}{\gamma^3}}}\right) \\
&= \sqrt{\frac{8T}{b}} \cdot L \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\| + \sqrt{\frac{2T}{b}} \cdot \frac{L}{\left\|\mathbf{w}_0 - \mathbf{w}_*\right\|} \cdot 8^{-\frac{1}{4}} c_2^{\frac{1}{2}} c \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|^2 \\
&\quad + \sqrt{\frac{2T}{b}} \cdot \frac{L}{\left\|\mathbf{w}_0 - \mathbf{w}_*\right\|} \cdot 8^{\frac{1}{4}} c_1^{\frac{1}{2}} c \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\| \times \\
&\quad \left(2\left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\| + 8^{\frac{1}{4}} c_1^{\frac{1}{2}} c \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\| + \sqrt{8^{-\frac{1}{4}} c_2^{\frac{1}{2}} c \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|^2}\right) \\
&= c_3 \sqrt{\frac{T}{b}} \cdot L \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|.
\end{aligned}$$

The suboptimality of $\widehat{\mathbf{w}}_T$ is then due to the convexity of $\phi(\mathbf{w})$:

$$\mathbb{E}\left[\phi(\widehat{\mathbf{w}}_T) - \phi(\mathbf{w}_*)\right] \le \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\phi(\tilde{\mathbf{w}}_t) - \phi(\mathbf{w}_*)\right] = \frac{c_3 L \left\|\tilde{\mathbf{w}}_0 - \mathbf{w}_*\right\|}{\sqrt{bT}}.$$

■

B.3. Proof of Theorem 8

Proof We have by Lemma 6 that

$$\mathbb{E}_{I_t,\mathcal{A}}\left[\phi(\tilde{\mathbf{w}}_t)-\phi(\mathbf{w}_*)\right] \le \frac{\lambda(t-1)}{4}\left\|\tilde{\mathbf{w}}_{t-1}-\mathbf{w}_*\right\|^2 - \frac{\lambda(t+1)}{4}\mathbb{E}_{I_t,\mathcal{A}}\left\|\tilde{\mathbf{w}}_t-\mathbf{w}_*\right\|^2 + \frac{8L^2}{\lambda b(t+1)} + \sqrt{\frac{4L^2\eta_t}{\lambda(t+1)}} + \sqrt{\lambda(t+1)\eta_t}\cdot\sqrt{\mathbb{E}_{I_t,\mathcal{A}}\left\|\tilde{\mathbf{w}}_t-\mathbf{w}_*\right\|^2}.$$

Relaxing the $\frac{1}{t+1}$ to $\frac{1}{t}$ on the RHS, and multiplying both sides by t, we further obtain

$$t\cdot\mathbb{E}_{I_t,\mathcal{A}}\left[\phi(\tilde{\mathbf{w}}_t)-\phi(\mathbf{w}_*)\right] \le \frac{\lambda(t-1)t}{4}\left\|\tilde{\mathbf{w}}_{t-1}-\mathbf{w}_*\right\|^2 - \frac{\lambda t(t+1)}{4}\mathbb{E}_{I_t,\mathcal{A}}\left\|\tilde{\mathbf{w}}_t-\mathbf{w}_*\right\|^2 + \frac{8L^2}{\lambda b} + \sqrt{\frac{4L^2 t\eta_t}{\lambda}} + \sqrt{\lambda t\eta_t}\cdot\sqrt{\mathbb{E}_{I_t,\mathcal{A}}\left[t(t+1)\left\|\tilde{\mathbf{w}}_t-\mathbf{w}_*\right\|^2\right]}.$$

Summing this inequality for $t=1,\dots,T$ yields

$$\begin{aligned}&\sum_{t=1}^T t\cdot\mathbb{E}\left[\phi(\tilde{\mathbf{w}}_t)-\phi(\mathbf{w}_*)\right] + \frac{\lambda T(T+1)}{4}\mathbb{E}\left\|\tilde{\mathbf{w}}_T-\mathbf{w}_*\right\|^2 \\ &\le \frac{8L^2T}{\lambda b} + \sum_{t=1}^T\sqrt{\frac{4L^2t\eta_t}{\lambda}} + \sum_{t=1}^T\sqrt{\lambda t\eta_t}\cdot\sqrt{\mathbb{E}\left[t(t+1)\left\|\tilde{\mathbf{w}}_t-\mathbf{w}_*\right\|^2\right]}. \qquad (25)\end{aligned}$$

Bounding $\sqrt{\mathbb{E}\left\|\tilde{\mathbf{w}}_t-\mathbf{w}_*\right\|^2}$. Dropping the $\sum_{t=1}^T t\cdot\mathbb{E}\left[\phi(\tilde{\mathbf{w}}_t)-\phi(\mathbf{w}_*)\right]$ term from (25) which is non-negative due to the optimality of $\mathbf{w}_*$, we obtain

$$\mathbb{E}\left[T(T+1)\left\|\tilde{\mathbf{w}}_T-\mathbf{w}_*\right\|^2\right] \le \frac{32L^2T}{\lambda^2 b} + \sum_{t=1}^T\sqrt{\frac{64L^2t\eta_t}{\lambda^3}} + \sum_{t=1}^T\sqrt{\frac{16t\eta_t}{\lambda}}\cdot\sqrt{\mathbb{E}\left[t(t+1)\left\|\tilde{\mathbf{w}}_t-\mathbf{w}_*\right\|^2\right]}.$$

Applying Lemma 12 (using $u_T=\sqrt{\mathbb{E}\left[T(T+1)\left\|\tilde{\mathbf{w}}_T-\mathbf{w}_*\right\|^2\right]}$, $S_T=\frac{32L^2T}{\lambda^2 b}+\sum_{t=1}^T\sqrt{\frac{64L^2t\eta_t}{\lambda^3}}$, and $\lambda_t=\sqrt{\frac{16t\eta_t}{\lambda}}$), we have

$$\sqrt{\mathbb{E}\left[T(T+1)\left\|\tilde{\mathbf{w}}_T-\mathbf{w}_*\right\|^2\right]} \le \sqrt{\frac{32L^2T}{\lambda^2 b}} + \sum_{t=1}^T\sqrt{\frac{16t\eta_t}{\lambda}} + \sqrt{\sum_{t=1}^T\sqrt{\frac{64L^2t\eta_t}{\lambda^3}}}.$$

Bounding function values. Dropping the $\mathbb{E}\left\|\tilde{\mathbf{w}}_T-\mathbf{w}_*\right\|^2$ term from (25) which is non-negative, we obtain

$$\begin{aligned}\sum_{t=1}^T t\cdot\mathbb{E}\left[\phi(\tilde{\mathbf{w}}_t)-\phi(\mathbf{w}_*)\right] &\le \frac{8L^2T}{\lambda b} + \sum_{t=1}^T\sqrt{\frac{4L^2t\eta_t}{\lambda}} \\ &+ \left(\sum_{t=1}^T\sqrt{\lambda t\eta_t}\right)\cdot\left(\sqrt{\frac{32L^2T}{\lambda^2 b}} + \sum_{t=1}^T\sqrt{\frac{16t\eta_t}{\lambda}} + \sqrt{\sum_{t=1}^T\sqrt{\frac{64L^2t\eta_t}{\lambda^3}}}\right). \qquad (26)\end{aligned}$$

To achieve the same order of regret as in exact minibatch-prox, we require that η_t decays with t, and in particular

$$\eta_t \le \min\left(c_1 \left(\frac{T}{b}\right),\, c_2 \left(\frac{T}{b}\right)^2 \right) \cdot \frac{L^2}{t^{3+2\delta}\lambda} \tag{27}$$

for some $\delta > 0$. Note that η_t has the unit of function value. Let $c := \sum_{i=1}^{\infty} \frac{1}{i^{1+\delta}} \le \frac{1+\delta}{\delta}$. Then (27) ensures that

$$\sum_{t=1}^{T} \sqrt{\frac{t\eta_t}{\lambda}} \le c\sqrt{c_1}\sqrt{\frac{L^2T}{\lambda^2 b}}, \qquad \text{and} \qquad \sum_{t=1}^{T} \sqrt{\frac{L^2 t\eta_t}{\lambda^3}} \le c\sqrt{c_2} \cdot \frac{L^2T}{\lambda^2 b}.$$

Continuing from (26), we have

$$\begin{aligned}\sum_{t=1}^{T} t \cdot \mathbb{E}\left[\phi(\tilde{\mathbf{w}}_t) - \phi(\mathbf{w}_*)\right] &\le \frac{8L^2T}{\lambda b} + 2c\sqrt{c_2} \cdot \frac{L^2T}{\lambda b} \\ &\quad + c\sqrt{c_1}\sqrt{\frac{L^2T}{b}} \left(\sqrt{\frac{32L^2T}{\lambda^2 b}} + 4c\sqrt{c_1}\sqrt{\frac{L^2T}{\lambda^2 b}} + \sqrt[4]{64c^2c_2}\sqrt{\frac{L^2T}{\lambda^2 b}} \right) \\ &= \frac{c_3}{2} \cdot \frac{L^2T}{\lambda b}.\end{aligned}$$

In view of the convexity of $\phi(\mathbf{w})$, by returning the weighted average $\widehat{\mathbf{w}}_T = \frac{2}{T(T+1)} \sum_{t=1}^{T} t\tilde{\mathbf{w}}_t$, we have

$$\mathbb{E}\left[\phi(\widehat{\mathbf{w}}_T) - \phi(\mathbf{w}_*)\right] \le \frac{2}{T(T+1)} \sum_{t=1}^{T} t \cdot \mathbb{E}\left[\phi(\tilde{\mathbf{w}}_t) - \phi(\mathbf{w}_*)\right] \le \frac{c_3 L^2}{\lambda b(T+1)}.$$

■

B.4. Connection to minibatch stochastic gradient descent

To see the connection between minibatch-prox and minibatch SGD, note that if we solve the linearized minibatch problem exactly, we obtain the minibatch stochastic gradient descent algorithm:

$$\tilde{\mathbf{w}}_t = \operatorname*{arg\,min}_{\mathbf{w}\in\Omega}\ \phi_{I_t}(\tilde{\mathbf{w}}_{t-1}) + \nabla \left\langle \phi_{I_t}(\tilde{\mathbf{w}}_{t-1}),\, \mathbf{w} - \tilde{\mathbf{w}}_{t-1} \right\rangle + \frac{\gamma_t}{2} \left\| \mathbf{w} - \tilde{\mathbf{w}}_{t-1} \right\|^2.$$

Following Cotter et al. (2011), we assume that $\ell(\mathbf{w}, \xi)$ is β-smooth:

$$\left\| \nabla\ell(\mathbf{w}, \xi) - \nabla\ell(\mathbf{w}', \xi) \right\| \le \beta \left\| \mathbf{w} - \mathbf{w}' \right\|, \qquad \forall \mathbf{w}, \mathbf{w}' \in \Omega.$$

We then have the following guarantee for each iterate of minbatch SGD.

Proposition 13 *For iteration t of minibatch SGD, we have*

$$\mathbb{E}_{I_t}\left[\phi(\tilde{\mathbf{w}}_t) - \phi(\mathbf{w}_*)\right] \le \frac{2L^2}{(\gamma_t - \beta)b} + \frac{\gamma_t - \lambda}{2} \left\| \mathbf{w}_* - \tilde{\mathbf{w}}_{t-1} \right\|^2 - \frac{\gamma_t}{2} \mathbb{E}_{I_t} \left\| \mathbf{w}_* - \tilde{\mathbf{w}}_t \right\|^2. \tag{28}$$

Proof Our proof closely follows that of Cotter et al. (2011).

Due to the smoothness of ϕ, we have that

$$\begin{aligned}
\phi(\tilde{\mathbf{w}}_t) &\le \phi(\tilde{\mathbf{w}}_{t-1}) + \langle \nabla\phi(\tilde{\mathbf{w}}_{t-1}),\, \tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\rangle + \frac{\beta}{2}\left\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\right\|^2 \\
&\le \phi(\tilde{\mathbf{w}}_{t-1}) + \langle \nabla\phi(\tilde{\mathbf{w}}_{t-1}) - \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1}),\, \tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\rangle + \frac{\beta}{2}\left\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\right\|^2 \\
&\qquad + \langle \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1}),\, \tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\rangle \\
&= \phi(\tilde{\mathbf{w}}_{t-1}) + \left\|\nabla\phi(\tilde{\mathbf{w}}_{t-1}) - \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1})\right\| \cdot \left\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\right\| + \frac{\beta}{2}\left\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\right\|^2 \\
&\qquad + \langle \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1}),\, \tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\rangle \\
&\le \phi(\tilde{\mathbf{w}}_{t-1}) + \frac{1}{2(\gamma_t-\beta)}\left\|\nabla\phi(\tilde{\mathbf{w}}_{t-1}) - \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1})\right\|^2 + \frac{\gamma_t-\beta}{2}\left\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\right\|^2 \\
&\qquad + \frac{\beta}{2}\left\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\right\|^2 + \langle \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1}),\, \tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\rangle \\
&= \phi(\tilde{\mathbf{w}}_{t-1}) + \frac{1}{2(\gamma_t-\beta)}\left\|\nabla\phi(\tilde{\mathbf{w}}_{t-1}) - \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1})\right\|^2 + \frac{\gamma_t}{2}\left\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\right\|^2 \\
&\qquad + \langle \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1}),\, \tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\rangle
\end{aligned} \tag{29}$$

where we have used the Cauchy-Schwarz inequality in the second inequality, and the inequality $xy \le \frac{x^2}{2\alpha} + \frac{\alpha y^2}{2}$ in the third inequality.

Now, since $\tilde{\mathbf{w}}_t$ is the minimizer of the γ_t-strongly convex function

$$\frac{\gamma_t}{2}\left\|\mathbf{w} - \tilde{\mathbf{w}}_{t-1}\right\|^2 + \langle \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1}),\, \mathbf{w} - \tilde{\mathbf{w}}_{t-1}\rangle$$

in Ω, we have according to Lemma 1 (replacing the local objective with its linear approximation) that

$$\begin{aligned}
&\frac{\gamma_t}{2}\left\|\mathbf{w}_* - \tilde{\mathbf{w}}_{t-1}\right\|^2 + \langle \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1}),\, \mathbf{w}_* - \tilde{\mathbf{w}}_{t-1}\rangle \\
&\quad \ge \frac{\gamma_t}{2}\left\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\right\|^2 + \langle \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1}),\, \tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{t-1}\rangle + \frac{\gamma_t}{2}\left\|\mathbf{w}_* - \tilde{\mathbf{w}}_t\right\|^2.
\end{aligned}$$

Substituting this into (29) gives

$$\begin{aligned}
\phi(\tilde{\mathbf{w}}_t) \le \phi(\tilde{\mathbf{w}}_{t-1}) &+ \frac{1}{2(\gamma_t-\beta)}\left\|\nabla\phi(\tilde{\mathbf{w}}_{t-1}) - \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1})\right\|^2 + \frac{\gamma_t}{2}\left\|\mathbf{w}_* - \tilde{\mathbf{w}}_{t-1}\right\|^2 \\
&+ \langle \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1}),\, \mathbf{w}_* - \tilde{\mathbf{w}}_{t-1}\rangle - \frac{\gamma_t}{2}\left\|\mathbf{w}_* - \tilde{\mathbf{w}}_t\right\|^2.
\end{aligned}$$

Taking expectation of this inequality over the random sampling of I_t further leads to

$$\begin{aligned}
\mathbb{E}_{I_t}\left[\phi(\tilde{\mathbf{w}}_t)\right] &\le \phi(\tilde{\mathbf{w}}_{t-1}) + \frac{1}{2(\gamma_t-\beta)}\mathbb{E}_{I_t}\left\|\nabla\phi(\tilde{\mathbf{w}}_{t-1}) - \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1})\right\|^2 + \frac{\gamma_t}{2}\left\|\mathbf{w}_* - \tilde{\mathbf{w}}_{t-1}\right\|^2 \\
&\qquad + \langle \nabla\phi(\tilde{\mathbf{w}}_{t-1}),\, \mathbf{w}_* - \tilde{\mathbf{w}}_{t-1}\rangle - \frac{\gamma_t}{2}\mathbb{E}_{I_t}\left\|\mathbf{w}_* - \tilde{\mathbf{w}}_t\right\|^2 \\
&\le \phi(\mathbf{w}_*) + \frac{1}{2(\gamma_t-\beta)}\mathbb{E}_{I_t}\left\|\nabla\phi(\tilde{\mathbf{w}}_{t-1}) - \nabla\phi_{I_t}(\tilde{\mathbf{w}}_{t-1})\right\|^2 \\
&\qquad + \frac{\gamma_t-\lambda}{2}\left\|\mathbf{w}_* - \tilde{\mathbf{w}}_{t-1}\right\|^2 - \frac{\gamma_t}{2}\mathbb{E}_{I_t}\left\|\mathbf{w}_* - \tilde{\mathbf{w}}_t\right\|^2
\end{aligned} \tag{30}$$

where in the second inequality we have used the fact that

$$\phi(\mathbf{w}_*) \geq \phi(\tilde{\mathbf{w}}_{t-1}) + \langle \nabla\phi(\tilde{\mathbf{w}}_{t-1}),\, \mathbf{w}_* - \tilde{\mathbf{w}}_{t-1} \rangle + \frac{\lambda}{2} \left\| \mathbf{w}_* - \tilde{\mathbf{w}}_{t-1} \right\|^2$$

due to the convexity of $\phi(\mathbf{w})$.

On the other hand, let $I_t = \{\xi_1, \dots, \xi_b\}$, we have

$$\begin{aligned}
& \mathbb{E}_{I_t} \left\| \nabla\phi(\mathbf{w}) - \nabla\phi_{I_t}(\mathbf{w}) \right\|^2 \\
&= \mathbb{E}_{I_t} \left\| \nabla\phi(\mathbf{w}) - \frac{1}{b}\sum_{i=1}^{b} \nabla\ell(\mathbf{w}, \xi_i) \right\|^2 \\
&= \mathbb{E}_{I_t} \left\| \frac{1}{b}\sum_{i=1}^{b} \left(\nabla\phi(\mathbf{w}) - \nabla\ell(\mathbf{w}, \xi_i) \right) \right\|^2 \\
&= \frac{1}{b^2}\sum_{i=1}^{b} \mathbb{E}_{\xi_i} \left\| \nabla\phi(\mathbf{w}) - \nabla\ell(\mathbf{w}, \xi_i) \right\|^2 + \frac{1}{b^2}\sum_{i\neq j} \mathbb{E}_{I_t} \left\langle \nabla\phi(\mathbf{w}) - \nabla\ell(\mathbf{w}, \xi_i),\, \nabla\phi(\mathbf{w}) - \nabla\ell(\mathbf{w}, \xi_j) \right\rangle \\
&= \frac{1}{b} \cdot \mathbb{E}_{\xi} \left\| \nabla\phi(\mathbf{w}) - \nabla\ell(\mathbf{w}, \xi) \right\|^2 \\
&\leq \frac{4L^2}{b}
\end{aligned}$$

where we used the fact that the samples are i.i.d. in the fourth equality, and that $\left\| \nabla\phi(\mathbf{w}) \right\|, \left\| \nabla\ell(\mathbf{w}, \xi) \right\| \leq L$ in the last inequality. Continuing from (30) yields the desired result. ■

Comparing this result to (20) and (21), we observe that minibatch SGD has a similar recursion to that exact minibatch-prox, except the appearance of β in the denominator of the "stability" term. We now show that this difference leads to significant difference in convergence rate.

Let $\ell(\mathbf{w}, \xi)$ be weakly convex ($\lambda = 0$), and $\gamma_t = \gamma$ for all $t \geq 1$. Summing (28) over $t = 1, \dots, T$ gives

$$\sum_{t=1}^{T} \mathbb{E}\left[\phi(\tilde{\mathbf{w}}_t) - \phi(\mathbf{w}_*) \right] \leq \frac{2L^2 T}{(\gamma - \beta) b} + \frac{\gamma}{2} \left\| \mathbf{w}_* - \tilde{\mathbf{w}}_0 \right\|^2 .$$

Minimizing the RHS over γ gives

$$\gamma = \beta + \sqrt{\frac{4T}{b}} \cdot \frac{L}{\left\| \mathbf{w}_* - \tilde{\mathbf{w}}_0 \right\|},$$

which leads to

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\phi(\tilde{\mathbf{w}}_t) - \phi(\mathbf{w}_*) \right] \leq \frac{2L \left\| \mathbf{w}_* - \tilde{\mathbf{w}}_0 \right\|}{\sqrt{bT}} + \frac{\beta \left\| \mathbf{w}_* - \tilde{\mathbf{w}}_0 \right\|^2}{2T}.$$

So we obtain the familiar $\mathcal{O}\left(\frac{1}{\sqrt{bT}} + \frac{1}{T} \right)$ rate for minibatch SGD.

Appendix C. Proof of Theorem 10

Proof On the one hand, as we choose γ as Theorem 7 suggested, we just need to verify that the inexactness conditions in Theorem 7 is satisfied, i.e., for $t = 1, \dots, T$, we require (recall that $\mathbf{w}_t^* = \arg\min_{\mathbf{w}} \ \tilde{f}_t(\mathbf{w})$)

$$\tilde{f}_t(\mathbf{w}_t) - \tilde{f}_t(\mathbf{w}_t^*) \le \frac{1}{10^4} \cdot \min\left(\left(\frac{T}{bm}\right)^{1/2}, \left(\frac{T}{bm}\right)^{3/2} \right) \cdot \frac{LB}{t^3}.$$

On the other hand, we can bound the initial suboptimality of $\tilde{f}_t(\mathbf{w})$ when initializing from $\mathbf{w}_{t-1}$. This is because, by the optimality of $\mathbf{w}_t^*$, we have $\|\mathbf{w}_t^* - \mathbf{w}_{t-1}\| = \left\| \frac{1}{\gamma} \nabla \phi_{I_t}(\mathbf{w}_t^*) \right\| \le L/\gamma$, and

$$\begin{aligned} \tilde{f}_t(\mathbf{w}_{t-1}) - \tilde{f}_t(\mathbf{w}_t^*) &= 0 + \phi_{I_t}(\mathbf{w}_{t-1}) - \frac{\gamma}{2} \|\mathbf{w}_t^* - \mathbf{w}_{t-1}\|^2 - \phi_{I_t}(\mathbf{w}_t^*) \\ &\le \phi_{I_t}(\mathbf{w}_{t-1}) - \phi_{I_t}(\mathbf{w}_t^*) \le L \|\mathbf{w}_t^* - \mathbf{w}_{t-1}\| \le L^2/\gamma. \end{aligned} \tag{31}$$

Combining the above two inequalities, the initial versus final error for the K DSVRG iterations is bounded by

$$\begin{aligned} & 10^4 \cdot \max\left(\left(\frac{bm}{T}\right)^{1/2}, \left(\frac{bm}{T}\right)^{3/2} \right) \cdot t^3 \cdot \frac{L}{B\gamma} \\ &= 10^4 \cdot \max\left(\left(\frac{bm}{T}\right)^{1/2}, \left(\frac{bm}{T}\right)^{3/2} \right) \cdot T^3 \cdot \frac{L}{B} \cdot \frac{bmB}{\sqrt{8n(\varepsilon)}L} \\ &= \mathcal{O}\left(\max\left(\frac{n(\varepsilon)^2}{bm}, bm \cdot n(\varepsilon) \right) \right) \\ &= \mathcal{O}\left(n^2(\varepsilon) \right) \end{aligned}$$

where we have used the definition of γ and $T = \frac{n(\varepsilon)}{bm}$ in the first and second step respectively.

By the iteration complexity results for sampling without-replacement DSVRG (Shamir, 2016, Theorem 4), we have the desired suboptimality in $\tilde{f}_t(\mathbf{w})$ using

$$K = \mathcal{O}\left(\log n(\varepsilon)\right) \tag{32}$$

iterations, as long as the batch size b/p_i is larger than the problem condition number.

Now, the condition number of $\tilde{f}(\mathbf{w})$ is

$$\frac{\beta + \gamma}{\gamma} = \mathcal{O}\left(\frac{\beta bmB}{\sqrt{n(\varepsilon)}L} \right).$$

Equating this to the batch size b/p_i yields the p_i specified in the theorem. It is also easy to check that $\frac{K}{\gamma} = \mathcal{O}(bm)$, i.e., the total number of stochastic updates is less than the total number of samples, as required by Shamir (2016, Theorem 4).

Communication: the total rounds of communication required by Algorithm 1 is

$$KT = \mathcal{O}\left(\frac{n(\varepsilon)}{mb} \log n(\varepsilon) \right).$$

Computation: For each communication round, each machine need to compute the local full gradient, which can be done in parallel, and then one of the machines perform b/p_i steps of stochastic update. So the computation cost is

$$KT\left(b+\frac{b}{p_i}\right)=\mathcal{O}\left(\frac{n(\varepsilon)}{m}\log n(\varepsilon)\right).$$

Memory: It is straightforward to see each machine only need to maintain b samples. ■

Appendix D. Communication-efficient distributed minibatch-prox with DANE

As discussed in Section 4, it is also possible to use other efficient distributed optimization solver for minibatch-prox. Here we present a novel method that use the distributed optimization algorithm DANE (Shamir et al., 2014) and its accelerated variant AIDE (Reddi et al., 2016) for solving (12), which define better local objectives than EMSO and take into consideration the similarity between local objectives.

We detail our algorithm, named MP-DANE, in Algorithm 2. The algorithm consists of three nested loops, where t, r and k are iteration counters for minibatch-prox (the outer **for**-loop), AIDE (the intermediate **for**-loop) and DANE (the inner **for**-loop) respectively. Compared to EMSO, DANE adds a gradient correction term to (13) which can be compute efficiently with one round of communication. On top of that, AIDE uses the idea of universal catalyst (Lin et al., 2015) and adds an extra quadratic term to improve the strong-convexity of the objective for faster convergence, i.e., in order to solve (12), AIDE solves multiple instances of the "augmented large minibatch" problems of the form

$$\min_{\mathbf{w}\in\Omega}\ \bar{f}_{t,r}(\mathbf{w}):=\phi_{I_t}(\mathbf{w})+\frac{\gamma}{2}\left\|\mathbf{w}-\mathbf{w}_{t-1}\right\|^2+\frac{\kappa}{2}\left\|\mathbf{w}-\mathbf{y}_{r-1}\right\|^2 \tag{36}$$

with carefully chosen extrapolation points $\mathbf{y}_{r-1}$. At each DANE iteration, we perform two rounds of communication, one for averaging the local gradients, and one for averaging the local updates, and the amount of data we communicate per round has the same size of the predictor.

To sum up, in Algorithm 2, we have introduced two levels of inexactness. First, we only approximately solve the "large minibatch" subproblem (12) in each outer loop; results from the previous section guarantee the convergence of this approach. Second, we only approximately solve the local subproblems (33) to sufficient accuracy in each inner loop; the analysis of "inexact DANE" (for the non-stochastic setting) provides guarantee for this approach (Reddi et al., 2016), and enables us to use state-of-the-art SGD methods (e.g., SVRG Johnson and Zhang, 2013; Xiao and Zhang, 2014) for solving local subproblems. Overall, we obtain a convergent algorithm for distributed stochastic convex optimization.

We now present detailed analysis for the computation/communication complexity of Algorithm 2 for stochastic quadratic problems, and compare it with related methods in the literature.

D.1. Efficiency of MP-DANE

We present the main results of this section (full analysis is deferred to Appendix D.3), which show that with careful choices of the minibatch size and the desired accuracy in each level of approxi-

Algorithm 2 MP-DANE for distributed stochastic convex optimization.

```
Initialize $\mathbf{w}_0$.
for $t = 1, 2, \ldots, T$ do
    Each machine $i$ draws a minibatch $I_t^{(i)}$ of $b$ samples from the underlying data distribution.
    Initialize $\mathbf{y}_0 \leftarrow \mathbf{w}_{t-1}, \quad \mathbf{x}_0 \leftarrow \mathbf{w}_{t-1}$.
    for $r = 1, 2, \ldots, R$ do
        Initialize $\mathbf{z}_0 \leftarrow \mathbf{y}_{r-1}$, $\alpha_0 = \sqrt{\gamma/(\gamma+\kappa)}$.
        for $k = 1, 2, \ldots, K$ do
            1. All machines perform one round of communication to compute the average gradient
               $$\nabla \phi_{I_t}(\mathbf{z}_{k-1}) \leftarrow \frac{1}{m} \sum_{i=1}^{m} \nabla \phi_{I_t^{(i)}}(\mathbf{z}_{k-1}).$$
            2. Each machine $i$ approximately solves the local objective to $\theta$-accuracy:
               $$\text{apply prox-SVRG to find } \mathbf{z}_k^{(i)} \quad \text{s.t.} \quad \left\| \mathbf{z}_k^{(i)} - \mathbf{z}_k^{(i)*} \right\| \le \theta \left\| \mathbf{z}_{k-1} - \mathbf{z}_k^{(i)*} \right\|$$
               $$\text{where } \mathbf{z}_k^{(i)^*} = \arg\min_{\mathbf{z}\in\Omega} \ \phi_{I_t^{(i)}}(\mathbf{z}) + \left\langle \nabla \phi_{I_t}(\mathbf{z}_{k-1}) - \nabla \phi_{I_t^{(i)}}(\mathbf{z}_{k-1}),\, \mathbf{z} \right\rangle + \frac{\gamma}{2} \left\| \mathbf{z} - \mathbf{w}_{t-1} \right\|^2 + \frac{\kappa}{2} \left\| \mathbf{z} - \mathbf{y}_{r-1} \right\|^2 . \quad (33)$$
            3. All machines reach consensus by averaging local updates through another round of communication:
               $$\mathbf{z}_k \leftarrow \frac{1}{m} \sum_{i=1}^{m} \mathbf{z}_k^{(i)}. \quad (34)$$
        end for
        Update $\mathbf{x}_r \leftarrow \mathbf{z}_K$.
        Compute $\alpha_r \in (0,1)$ such that $\alpha_r^2 = (1-\alpha_r)\alpha_{r-1}^2 + \gamma\alpha_k/(\gamma+\kappa)$, and compute
        $$\mathbf{y}_r = \mathbf{x}_r + \left( \frac{\alpha_{r-1}(1-\alpha_{r-1})}{\alpha_r + \alpha_{r-1}^2} \right) (\mathbf{x}_r - \mathbf{x}_{r-1}). \quad (35)$$
    end for
    Update $\mathbf{w}_t \leftarrow \mathbf{x}_r$.
end for
Output: $\mathbf{w}_T$ is the approximate solution.
```

mate solution, MP-DANE achieves both communication and computation efficiency with the optimal sample complexity. Interestingly, the choices of parameters differ in two regimes which are separated by an "optimal" minibatch size (also denoted as $b_{\text{mp-dane}}$ in the main text)

$$b^* = \frac{n(\varepsilon)L^2}{32 m^2 \beta^2 B^2 \log(md)}.$$

Theorem 14 (Efficiency of MP-DANE for $b \le b^*$) *Set the parameters in Algorithm 2 as follows:*

$$\text{(outer loop)} \qquad b \le b^* = \frac{n(\varepsilon)L^2}{32m^2\beta^2B^2\log(md)}, \quad T = \frac{n(\varepsilon)}{bm}, \quad \gamma = \frac{\sqrt{8n(\varepsilon)}L}{bmB},$$
$$\text{(intermediate loop)} \qquad \kappa = 0, \quad R = 1,$$
$$\text{(inner loop)} \qquad \theta = \frac{1}{6}, \quad K = \mathcal{O}\left(\log n(\varepsilon)\right).$$

Then we have $\mathbb{E}\left[\phi\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t\right) - \phi(\mathbf{w}_*)\right] \le \frac{\sqrt{40}BL}{\sqrt{n(\varepsilon)}} = \mathcal{O}\left(\varepsilon\right).$

Moreover, Algorithm 2 can be implemented with $\tilde{\mathcal{O}}\left(\frac{n(\varepsilon)}{bm}\right)$ *rounds of communication, and each machine performs* $\tilde{\mathcal{O}}\left(\frac{n(\varepsilon)}{m}\right)$ *vector operations in total, where the notation* $\tilde{\mathcal{O}}(\cdot)$ *hides poly-logarithmic dependences on* $n(\varepsilon)$.

When we choose $b = b^*$, *Algorithm 1 can be implemented with* $\tilde{\mathcal{O}}\left(\frac{m\beta^2B^2}{L^2}\right)$ *rounds of communication,* $\tilde{\mathcal{O}}\left(\frac{n(\varepsilon)}{bm}\right)$ *vector operations, and* $\mathcal{O}\left(\frac{n(\varepsilon)L^2}{m^2\beta^2B^2}\right)$ *memory for each machine.*

We comment on the choice of parameters. For sample efficiency, we fix the sample size $n(\varepsilon)$ and number of machines m, and so we can tradeoff the local minibatch size b and the total number of outer iterations T, maintaining $bT = \frac{n(\varepsilon)}{m}$. For any b, the regularization parameters in the "large minibatch" problem is set to $\gamma = \sqrt{\frac{8T}{bm}} \cdot \frac{L}{B} = \frac{\sqrt{8n(\varepsilon)}L}{bmB}$ according to Theorem 7. When $b \le b^*$, we note that (37) can be satisfied with $\kappa = 0$ and there is no need for acceleration by AIDE ($R = 1$). Then the values of θ and K follow from Lemma 18.

Remark 15 *The above theorem suggests that in the regime of* $b \le b^*$, *we only need to have logarithmic number of DANE iterations for solving each "large minibatch" problem, and logarithmic number of passes over the local data during each DANE iteration. We present experimental results validating our theory in Appendix E.*

The next theorem shows that when we use a large minibatch size b in Algorithm 2, we can still satisfy the condition (37) by adding extra regularization ($\kappa > 0$), and then apply accelerated DANE.

Theorem 16 (Efficiency of MP-DANE for $b \ge b^*$) *Set the parameters in Algorithm 2 as follows:*

$$\text{(outer loop)} \qquad b \ge b^* = \frac{n(\varepsilon)L^2}{32m^2\beta^2B^2\log(md)}, \quad T = \frac{n(\varepsilon)}{bm}, \quad \gamma = \frac{\sqrt{8n(\varepsilon)}L}{bmB},$$
$$\text{(intermediate loop)} \qquad \kappa = 16\beta\sqrt{\frac{\log(dm)}{b}} - \gamma, \quad R = \mathcal{O}\left(\frac{b^{1/4}m^{1/2}\cdot\beta^{1/2}B^{1/2}}{n(\varepsilon)^{1/4}\cdot L^{1/2}}\log n(\varepsilon)\right),$$
$$\text{(inner loop)} \qquad \theta = \frac{1}{6}, \quad K = \mathcal{O}\left(\log n(\varepsilon)\right).$$

Then we have $\mathbb{E}\left[\phi\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t\right) - \phi(\mathbf{w}_*)\right] \le \frac{\sqrt{40}BL}{\sqrt{n(\varepsilon)}} = \mathcal{O}\left(\varepsilon\right).$

Moreover, Algorithm 2 can be implemented with $\tilde{\mathcal{O}}\left(\frac{n(\varepsilon)^{3/4}\cdot\beta^{1/2}B^{1/2}}{b^{3/4}m^{1/2}\cdot L^{1/2}}\right)$ *rounds of communication, and each machine performs* $\tilde{\mathcal{O}}\left(\frac{b^{1/4}n(\varepsilon)^{3/4}\cdot\beta^{1/2}B^{1/2}}{m^{1/2}\cdot L^{1/2}}\right)$ *vector operations in total, where the notation* $\tilde{\mathcal{O}}(\cdot)$ *hides poly-logarithmic dependences on* $n(\varepsilon)$.

	Samples	Communication	Computation	Memory
$1 \le b \le b^*$	$n(\varepsilon)$	$n(\varepsilon)/mb$	$n(\varepsilon)/m$	b
$b = b^*$	$n(\varepsilon)$	$B^2 m$	$n(\varepsilon)/m$	$n(\varepsilon)/(m^2B^2)$
$b^* < b \le b_{\max}$	$n(\varepsilon)$	$B^{1/2}n(\varepsilon)^{3/4}/(m^{1/2}b^{3/4})$	$B^{1/2}n(\varepsilon)^{3/4}b^{1/4}/m^{1/2}$	b

Table 2: Summary of resources required by MP-DANE for distributed stochastic convex optimization, in units of vector operations/communications/memory per machine, ignoring constants and log-factors. Here $b^* \asymp n(\varepsilon)/(m^2B^2)$, and $b_{\max} = n(\varepsilon)/m$.

D.2. Two regimes of multiple resource tradeoffs

From the above analysis, we summarized in Table 2 the resources required by MP-DANE. We observe two interesting regimes, separated by the minibatch size $b^* \asymp n(\varepsilon)/(m^2B^2)$, that present different tradeoffs between communication, computation and memory.

- When $1 \le b \le b^*$, the computation complexity remains $\tilde{\mathcal{O}}\left(n(\varepsilon)/m\right)$ which is independent of b. This means we always achieve *near-linear speedup* in this regime. Moreover, there is a tradeoff between communication and memory: the communication complexity decreases, while the memory cost increases as the minibatch size b increases, both at the linear rate. Thus in this regime, we can trade communication for memory without affecting computation.
- When $b^* < b \le b_{\max}$, the computation starts to increase with b at the rate $b^{1/4}$ which is slower than linear, while the communication cost continues to decrease at the rate $b^{3/4}$ which is also slower than linear. Thus in this regime, we can trade communication for computation and memory.

D.3. Analysis of MP-DANE

In order to fully analyze Algorithm 2, we need several auxiliary lemmas that characterize the iteration complexity of solving the local problem (33) by prox-SVRG (Xiao and Zhang, 2014), the large minibatch problem (12) by DANE (Shamir et al., 2014) and AIDE (Reddi et al., 2016).

D.3.1. SOME AUXILIARY LEMMAS

First, we apply prox-SVRG to the local problem (33), pushing all terms but $\phi_{I_t^{(i)}}(\mathbf{z})$ in to the proximal operator. The benefit of this approach (as opposed to using plain SVRG Johnson and Zhang, 2013) is that the smoothness parameter that determines the iteration complexity is simply β, same results hold when applying prox-SAGA (Defazio et al., 2014) as well. For sampling without replacement SVRG, the current analysis works only for plain SVRG, so we quote the results from (Shamir, 2016).

Lemma 17 (Iteration complexity of SVRG for (33)) *For any target accuracy $\theta > 0$, with initialization $\mathbf{z}_{k-1}$, prox-SVRG outputs $\mathbf{z}_k^{(i)}$ such that $\left\|\mathbf{z}_k^{(i)} - \mathbf{z}_k^{(i)*}\right\| \le \theta \left\|\mathbf{z}_{k-1} - \mathbf{z}_k^{(i)*}\right\|$ after*

$$\mathcal{O}\left(\left(b + \frac{\beta}{\gamma+\kappa}\right) \cdot \log \frac{(\beta+\gamma+\kappa)}{(\gamma+\kappa)\theta^2}\right)$$

vector operations, and sampling without replacement SVRG outputs $\mathbf{z}_k^{(i)}$ *such that* $\left\|\mathbf{z}_k^{(i)} - \mathbf{z}_k^{(i)*}\right\| \le \theta \left\|\mathbf{z}_{k-1} - \mathbf{z}_k^{(i)*}\right\|$ *after*

$$\mathcal{O}\left(\left(b + \frac{\beta+\kappa}{\gamma+\kappa}\right) \cdot \log \frac{(\beta+\gamma+\kappa)}{(\gamma+\kappa)\theta^2}\right)$$

vector operations.

Proof Observe that the objective (33) by $f_k^{(i)}(\mathbf{z})$, which is an quadratic function of $\mathbf{z}$ with the Hessian matrix $H_i = \nabla^2 \phi_{I_t^{(i)}}(\mathbf{z}) + (\gamma+\kappa)\mathbf{I} \succeq (\gamma+\kappa)\mathbf{I}$. As a result, the suboptimality of $\mathbf{z}_k^{(i)}$ is

$$\epsilon_{\text{final}} = f_k^{(i)}(\mathbf{z}_k^{(i)}) - f_k^{(i)}(\mathbf{z}_k^{(i)*}) = \frac{1}{2}\left(\mathbf{z}_k^{(i)} - \mathbf{z}_k^{(i)*}\right)^\top H_i \left(\mathbf{z}_k^{(i)} - \mathbf{z}_k^{(i)*}\right) \ge \frac{\gamma+\kappa}{2}\left\|\mathbf{z}_k^{(i)} - \mathbf{z}_k^{(i)*}\right\|^2.$$

To satisfy the requirement of $\left\|\mathbf{z}_k^{(i)} - \mathbf{z}_k^{(i)*}\right\| \le \theta \left\|\mathbf{z}_{k-1} - \mathbf{z}_k^{(i)*}\right\|$, we require

$$\epsilon_{\text{final}} \le \frac{(\gamma+\kappa)\theta^2}{2}\left\|\mathbf{z}_{k-1} - \mathbf{z}_k^{(i)*}\right\|^2.$$

On the other hand, when initializing from z_{k-1}, the initial suboptimality is

$$\epsilon_{\text{init}} = f_k^{(i)}(\mathbf{z}_{k-1}) - f_k^{(i)}(\mathbf{z}_k^{(i)*}) \le \frac{\sigma_{\max}(H_i)}{2}\left\|\mathbf{z}_{k-1} - \mathbf{z}_k^{(i)*}\right\|^2 \le \frac{\beta+\gamma+\kappa}{2}\left\|\mathbf{z}_{k-1} - \mathbf{z}_k^{(i)*}\right\|^2.$$

Therefore, it suffices to have

$$\frac{\epsilon_{\text{init}}}{\epsilon_{\text{final}}} = \frac{(\beta+\gamma+\kappa)}{(\gamma+\kappa)\theta^2}.$$

Noting that $\phi_{I_t^{(i)}}(\mathbf{z})$ is the sum of b components, and each component is β-smooth while the overall function $f_k^{(i)}$ is $(\gamma+\kappa)$-strongly convex, the lemma follows directly from the convergence guarantee of prox-SVRG (Xiao and Zhang, 2014, Corollary 1), and sampling without replacement SVRG (Shamir, 2016, Theorem 4). ■

Next, we state the convergence rates of "inexact DANE" and AIDE, which can be easily derived from Reddi et al. (2016). At the outer loop t and intermediate loop r, let $\mathbf{x}_r^* = \arg\min_{\mathbf{w}} \bar{f}_{t,r}(\mathbf{w})$ be the exact minimizer of the "augmented large minibatch" problem (36), which is approximately solved by the inner DANE iterations.

Lemma 18 (Iteration Complexity of inexact DANE) *Let* $\theta = \frac{1}{6}$*, and assume that*

$$b(\gamma+\kappa)^2 \ge 256\beta^2 \log(dm/\delta). \tag{37}$$

By initializing from $\mathbf{y}_{r-1}$*, and setting the number of inner iterations in Algorithm 2 to be*

$$K = \left\lceil \frac{1}{2}\log_{4/3} \frac{(\beta+\gamma+\kappa)}{(\gamma+\kappa)\eta} \right\rceil,$$

we have with probability $1-\delta$ *over the sample set* I_t *that*

$$\bar{f}_{t,r}(\mathbf{x}_r) - \bar{f}_{t,r}(\mathbf{x}_r^*) \le \eta\left(\bar{f}_{t,r}(\mathbf{y}_{r-1}) - \bar{f}_{t,r}(\mathbf{x}_r^*)\right).$$

Proof Denote by $H_i = \nabla^2 \phi_{I_t^{(i)}}(\mathbf{z}) + (\gamma + \kappa)\mathbf{I}$ the Hessian matrix of the local objective (33) for machine i. Let $H = \frac{1}{m}\sum_{i=1}^m H_i$ be the Hessian matrix of the global objective (36), and $\tilde{H}^{-1} = \frac{1}{m}\sum_{i=1}^m H_i^{-1}$. As our objective is quadratic, $H_i, H, \tilde{H}^{-1}$ remain unchanged during the inner iterations. By Reddi et al. (2016, Theorem 1), we have

$$\|\mathbf{z}_k - \mathbf{x}_r^*\| \le \left(\left\| \tilde{H}^{-1} H - \mathbf{I} \right\| + \frac{\theta}{m} \sum_{i=1}^m \left\| H_i^{-1} H \right\| \right) \|\mathbf{z}_{k-1} - \mathbf{x}_r^*\| . \tag{38}$$

Since $\nabla^2 \ell(\mathbf{w}, \xi) \le \beta$, by Shamir et al. (2014, Lemma 2), we have with probability at least $1 - \delta$ over the sample set I_t that

$$\|H_i - H\| \le \sqrt{\frac{32\beta^2 \log(dm/\delta)}{b}} =: \rho, \qquad i = 1, \dots, m.$$

On the other hand, we have $H_i \succeq (\gamma + \kappa)\mathbf{I}$ and

$$\frac{4\rho^2}{(\gamma+\kappa)^2} = \frac{128\beta^2 \log(dm/\delta)}{b(\gamma+\kappa)^2} \le \frac{1}{2}$$

by our assumption (37). By Shamir et al. (2014, Lemma 1), we have

$$\left\| \tilde{H}^{-1} H - \mathbf{I} \right\| \le \frac{1}{2}. \tag{39}$$

Moreover, we have

$$\begin{aligned}
\frac{\theta}{m} \sum_{i=1}^m \left\| H_i^{-1} H \right\| &\le \frac{\theta}{m} \sum_{i=1}^m \left(1 + \left\| H_i^{-1} H - \mathbf{I} \right\|\right) \\
&\le \frac{\theta}{m} \sum_{i=1}^m \left(1 + \left\| H_i^{-1} \right\| \left\| H - H_i^{-1} \right\|\right) \\
&\le \frac{\theta}{m} \sum_{i=1}^m \left(1 + \frac{\rho}{\gamma+\kappa}\right) \\
&\le \frac{\theta}{m} \sum_{i=1}^m \left(1 + \frac{1}{2\sqrt{2}}\right) \\
&\le \frac{3\theta}{2} \le \frac{1}{4}.
\end{aligned} \tag{40}$$

Plugging (39) and (40) into (38) yields

$$\|\mathbf{z}_k - \mathbf{x}_r^*\| \le \frac{3}{4} \|\mathbf{z}_{k-1} - \mathbf{x}_r^*\| ,$$

and thus $\|\mathbf{z}_K - \mathbf{x}_r^*\| \le (3/4)^K \|\mathbf{y}_{r-1} - \mathbf{x}_r^*\|$. To guarantee the suboptimality in the objective $\bar{f}_{t,r}(\mathbf{w})$, we note that

$$\begin{aligned}\bar{f}_{t,r}(\mathbf{z}_K) - \bar{f}_{t,r}(\mathbf{x}_r^*) &= \frac{1}{2}(\mathbf{z}_K - \mathbf{x}_r^*)^\top H(\mathbf{z}_K - \mathbf{x}_r^*) \le \frac{\beta+\gamma+\kappa}{2}\|\mathbf{z}_K - \mathbf{x}_r^*\|^2 \\ &\le \left(\frac{3}{4}\right)^{2K}\frac{\beta+\gamma+\kappa}{2}\|\mathbf{y}_{r-1} - \mathbf{x}_r^*\|^2 \\ &\le \left(\frac{3}{4}\right)^{2K}\frac{\beta+\gamma+\kappa}{\gamma+\kappa}\left(\bar{f}_{t,r}(\mathbf{y}_{r-1}) - \bar{f}_{t,r}(\mathbf{x}_r^*)\right)\end{aligned}$$

where we have used the fact that $f_{t,r}(\mathbf{w})$ is $(\gamma+\kappa)$-strongly convex in the last inequality. Setting $\left(\frac{3}{4}\right)^{2K}\frac{\beta+\gamma+\kappa}{\gamma+\kappa} = \eta$, and noting $\mathbf{x}_r = \mathbf{z}_K$, we obtain the desired iteration complexity. ■

At the outer iteration t of Algorithm 2, we are trying to approximately minimize the objective (12) by iteratively (approximately) solving R instances of the "augmented" problem (36). Let $\mathbf{w}_t^*$ be the exact minimizer of the "large minibatch" subproblem (12):

$$\mathbf{w}_t^* = \arg\min_{\mathbf{w}} \ \tilde{f}_t(\mathbf{w}).$$

The following lemma characterizes the accelerated convergence rate.

Lemma 19 (Acceleration by universal catalyst, Theorem 3.1 of Lin et al. (2015)) *Assume that for all $r \ge 1$, we have*

$$\bar{f}_{t,r}(\mathbf{x}_r) - \bar{f}_{t,r}(\mathbf{x}_r^*) \le \frac{2}{9}\left(1 - \frac{9}{10}\sqrt{\frac{\gamma}{\gamma+\kappa}}\right)^R \cdot \left(\tilde{f}_t(\mathbf{x}_0) - \tilde{f}_t(\mathbf{w}_t^*)\right),$$

then

$$\tilde{f}_t(\mathbf{x}_R) - \tilde{f}_t(\mathbf{w}_t^*) \le \frac{800(\gamma+\kappa)}{\gamma}\left(1 - \frac{9}{10}\sqrt{\frac{\gamma}{\gamma+\kappa}}\right)^{R+1}\left(\tilde{f}_t(\mathbf{x}_0) - \tilde{f}_t(\mathbf{w}_t^*)\right).$$

D.3.2. PROOF OF THEOREM 14

Proof First of all, because $R = 1$, our algorithm collapses into two nested loops.

On the one hand, as we choose γ as Theorem 7 suggested, we just need to verify the inexactness conditions in Theorem 7 is satisfied, i.e., for $t = 1, \dots, T$, we require (recall that $\mathbf{w}_t^* = \arg\min_{\mathbf{w}} \ \tilde{f}_t(\mathbf{w})$)

$$\tilde{f}_t(\mathbf{w}_t) - \tilde{f}_t(\mathbf{w}_t^*) \le \frac{1}{10^4} \cdot \min\left(\left(\frac{T}{bm}\right)^{1/2}, \left(\frac{T}{bm}\right)^{3/2}\right) \cdot \frac{2LB}{t^3}.$$

On the other hand, we can bound the initial suboptimality $\tilde{f}_t(\mathbf{w})$ (cf. derivation for (31)):

$$\tilde{f}_t(\tilde{\mathbf{w}}_{t-1}) - \tilde{f}_t(\mathbf{w}_t^*) \le L^2/\gamma.$$

Using Lemma 18, we know as long as the inequality (37) is satisfied, we have the desired suboptimality in $\tilde{f}_t(\mathbf{w})$ using (cf. the derivation for (32))

$$K = \mathcal{O}\left(\log n(\varepsilon)\right)$$

rounds of communication, where we have plugged in the value of γ in the second step.

It remains to verify the condition (37), by our choice of γ and b, we have

$$b\gamma^2 = \frac{8n(\varepsilon)L^2}{bm^2B^2} \geq \frac{8n(\varepsilon)L^2}{b^*m^2B^2} = 256\beta^2\log(md), \tag{41}$$

as desired.

Next we summarize the communication, computation, and memory efficiency.

Communication: the total rounds of communication required by Algorithm 2 is

$$KRT = \mathcal{O}\left(\frac{n(\varepsilon)}{mb}\log n(\varepsilon)\right).$$

Computation: For each communication round, we need to solve the local problem (33) using prox-SVRG. Now, in view of (41), we have $\beta = \mathcal{O}(\sqrt{b}\gamma)$. This implies that $\frac{\beta}{\gamma} = \mathcal{O}(\sqrt{b})$ and thus by Lemma 17, the dominant term of the iteration complexity of prox-SVRG is

$$\mathcal{O}\left(b\log\frac{\beta+\gamma}{\gamma}\right) = \mathcal{O}\left(b\log n(\varepsilon)\right).$$

Multiplying this with the number of communication rounds yields the desired computation complexity.

Memory: It is straightforward to see each machine only need to maintain b samples. ■

D.3.3. PROOF OF THEOREM 16

Proof First, it is straightforward to verify the condition (37):

$$b(\gamma+\kappa)^2 = 256\beta^2\log(dm).$$

Similarly to Theorem 14, we need the ratio between final versus initial error for the R AIDE iterations to be

$$\text{ratio} = \mathcal{O}(n(\varepsilon)).$$

Equating this ratio to be $\frac{800(\gamma+\kappa)}{\gamma}\left(1-\frac{9}{10}\sqrt{\frac{\gamma}{\gamma+\kappa}}\right)^{R+1}$, we have

$$\begin{aligned} R &= \frac{10}{9}\sqrt{\frac{\gamma+\kappa}{\gamma}}\log\left(\frac{800(\gamma+\kappa)}{\gamma}\cdot\frac{1}{ratio}\right) \\ &= \mathcal{O}\left(\frac{b^{1/4}m^{1/2}\cdot\beta^{1/2}B^{1/2}}{n(\varepsilon)^{1/4}\cdot L^{1/2}}\log n(\varepsilon)\right). \end{aligned}$$

Now according to Lemma 19, the final suboptimality for $\bar{f}_{t,r}(\mathbf{w})$ need to be

$$\epsilon_{\text{final}} = \frac{2}{9}\left(1-\frac{9}{10}\sqrt{\frac{\gamma}{\gamma+\kappa}}\right)^R\cdot\left(\tilde{f}_t(\mathbf{x}_0)-\tilde{f}_t(\mathbf{w}_t^*)\right).$$

Let us initialize $\min_{\mathbf{w}} \ \bar{f}_{t,r}(\mathbf{w})$ by $\mathbf{x}_0$. By definition, we have $\bar{f}_{t,r}(\mathbf{w}) \geq \tilde{f}_t(\mathbf{w})$ and thus

$$\begin{aligned}\epsilon_{\text{init}} &= \bar{f}_{t,r}(\mathbf{x}_0) - \bar{f}_{t,r}(\mathbf{x}_r^*) \\ &\leq \tilde{f}_t(\mathbf{x}_0) - \tilde{f}_t(\mathbf{x}_r*) \\ &\leq \tilde{f}_t(\mathbf{x}_0) - \tilde{f}_t(\mathbf{w}_t^*)\end{aligned}$$

where we have used the fact that $\mathbf{w}_t^*$ is the minimizer of $\tilde{f}_t(\mathbf{w})$ in the second inequality.

This means we only need the initial versus final suboptimality of solving $\bar{f}_{t,r}(\mathbf{w})$ to be

$$\frac{1}{\eta} = \frac{\epsilon_{\text{init}}}{\epsilon_{\text{final}}} = \frac{9}{2}\left(1 - \frac{9}{10}\sqrt{\frac{\gamma}{\gamma+\kappa}}\right)^{-R},$$

which, according to Lemma 18, is achieved by inexact DANE with

$$\begin{aligned}K &= \mathcal{O}\left(\log\frac{1}{\eta} + \log\frac{\beta+\gamma+\kappa}{\gamma+\kappa}\right) \\ &= \mathcal{O}\left(R\sqrt{\frac{\gamma}{\gamma+\kappa}}\right) \\ &= \mathcal{O}\left(\log n(\varepsilon)\right).\end{aligned}$$

iterations.

Next we analyze the communication and computation efficiency of our algorithm.

Communication: The total rounds of communication is

$$\begin{aligned}KRT &= \mathcal{O}\left(\log n(\varepsilon) \cdot \frac{b^{1/4}m^{1/2} \cdot \beta^{1/2}B^{1/2}}{n(\varepsilon)^{1/4} \cdot L^{1/2}} \log n(\varepsilon) \cdot \frac{n(\varepsilon)}{bm}\right) \\ &= \mathcal{O}\left(\frac{n(\varepsilon)^{3/4} \cdot \beta^{1/2}B^{1/2}}{b^{3/4}m^{1/2} \cdot L^{1/2}} \log^2 n(\varepsilon)\right).\end{aligned}$$

Computation: Similar to the case of $b \leq b^*$, for each DANE local subproblem (33), the sample size b is larger than its condition number. Therefore, the total computational cost is

$$\mathcal{O}(bKRT) = \mathcal{O}\left(\frac{b^{1/4}n(\varepsilon)^{3/4} \cdot \beta^{1/2}B^{1/2}}{m^{1/2} \cdot L^{1/2}} \log^2 n(\varepsilon)\right).$$

■

Appendix E. Experiments

In this section we present empirical results to support our theoretical analysis of MP-DANE. We perform least squares regression and classification on several publicly available datasets[6]; the statistics of these datasets and the corresponding losses are summarized in Table 3. For each dataset, we

6. https://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 3: List of datasets used in the experiments.

Name	#Samples	#Features	loss
codrna	271,617	8	logistic
covtype	581,012	54	logistic
kddcup99	1,131,571	127	logistic
year	463,715	90	squared

randomly select half of the samples for training, and the remaining samples are used for estimating the stochastic objective.

For MP-DANE, we use SAGA (Defazio et al., 2014) to solve each local DANE subproblem (33) and fix the number of SAGA steps to b (i.e., we just make one pass over the local data), while varying the number of DANE rounds K over $\{1, 2, 4, 8, 16\}$. For simplicity, we do not use catalyst acceleration and set $R = 1$ and $\kappa = 0$ in all experiments. Our experiments simulate a distributed environment with m machines, for $m = 4, 8, 16$. We conduct a simple comparison with minibatch SGD. Stepsizes for SAGA and minibatch SGD are set based on the smoothness parameter of the loss.

We plot in Figure 3 the estimated population objective vs. minibatch size b for different parameters. We make the following observations.

- For minibatch SGD, as b increases, the objective often increases quickly, this is because minibatch SGD can not uses large minibatch sizes while preserving sample efficiency.
- For MP-DANE, the objective increases much more slowly as b increases. This demonstrates the effectiveness of minibatch-prox for using large minibatch sizes.
- Running more iterations of DANE often helps, but with diminishing returns. This validates our theory that only a near-constant number of DANE iterations is needed for solving the large minibatch objective, without affecting the sample efficiency.

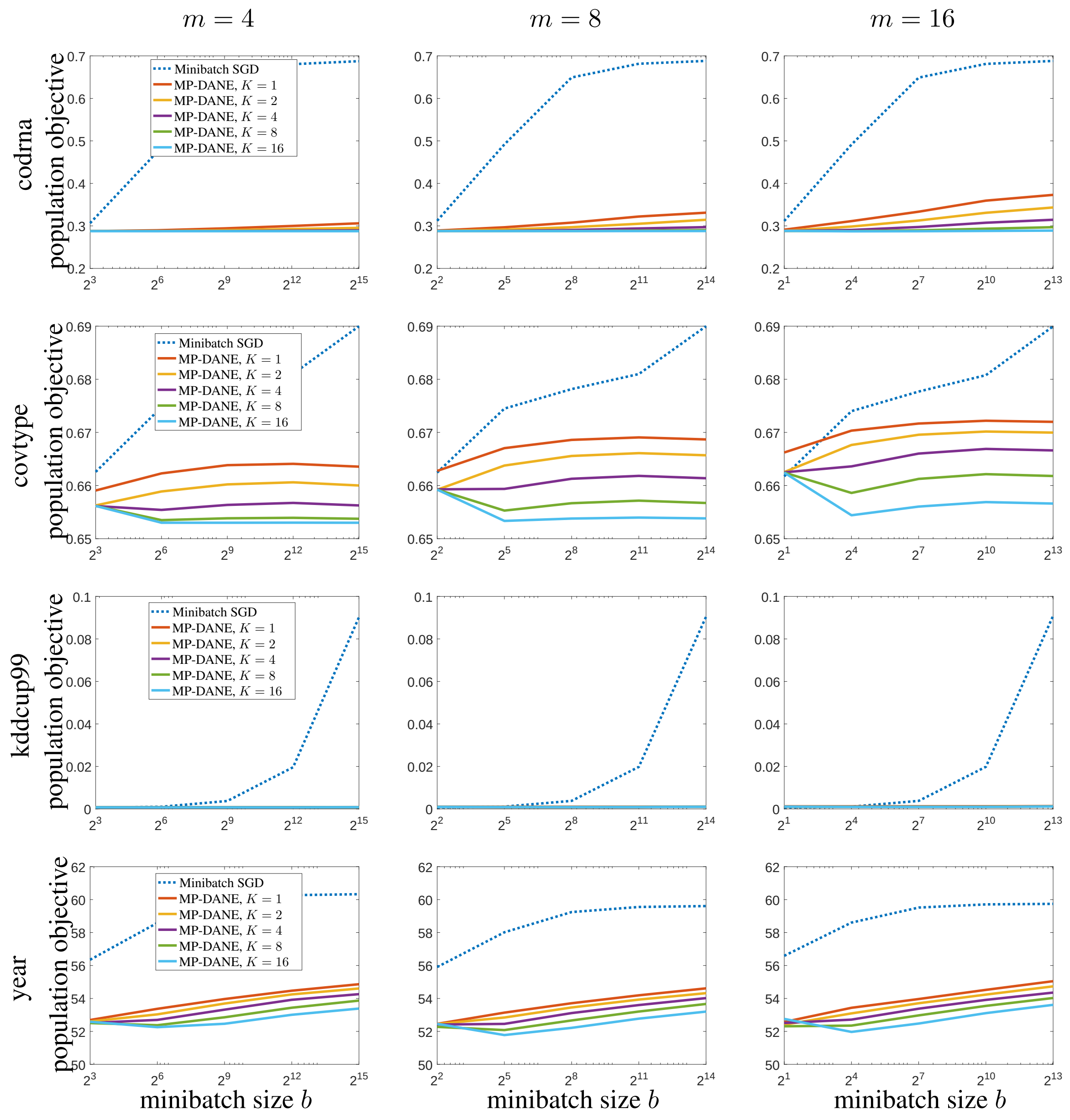

Figure 3: Illustration of the convergence properties of MP-DANE, for different minibatch size b, number of machines m, and number of DANE iterations K.