Hybrid Stochastic-Deterministic Minibatch Proximal Gradient: Less-Than-Single-Pass Optimization with Nearly Optimal Generalization

Pan Zhou [1] **Xiao-Tong Yuan** [2]

Abstract

Stochastic variance-reduced gradient (SVRG) algorithms have been shown to work favorably in solving large-scale learning problems. Despite the remarkable success, the stochastic gradient complexity of SVRG-type algorithms usually scales linearly with data size and thus could still be expensive for huge data. To address this deficiency, we propose a hybrid stochastic-deterministic minibatch proximal gradient (HSDMPG) algorithm for strongly-convex problems that enjoys provably improved data-size-independent complexity guarantees. More precisely, for quadratic loss $F(\boldsymbol{\theta})$ of n components, we prove that HSDMPG can attain an ϵ-optimization-error $\mathbb{E}[F(\boldsymbol{\theta})-F(\boldsymbol{\theta}^*)] \leq \epsilon$ within $\mathcal{O}\Big(\frac{\kappa^{1.5}\epsilon^{0.75}\log^{1.5}(\frac{1}{\epsilon})+1}{\epsilon} \wedge \Big(\kappa\sqrt{n}\log^{1.5}\big(\frac{1}{\epsilon}\big)+ n\log\big(\frac{1}{\epsilon}\big)\Big)\Big)$ stochastic gradient evaluations, where κ is condition number. For generic strongly convex loss functions, we prove a nearly identical complexity bound though at the cost of slightly increased logarithmic factors. For large-scale learning problems, our complexity bounds are superior to those of the prior state-of-the-art SVRG algorithms with or without dependence on data size. Particularly, in the case of $\epsilon = \mathcal{O}\big(1/\sqrt{n}\big)$ which is at the order of intrinsic excess error bound of a learning model and thus sufficient for generalization, the stochastic gradient complexity bounds of HSDMPG for quadratic and generic loss functions are respectively $\mathcal{O}(n^{0.875}\log^{1.5}(n))$ and $\mathcal{O}(n^{0.875}\log^{2.25}(n))$, which to our best knowledge, for the first time achieve optimal generalization in less than a single pass over data. Extensive numerical results demonstrate the computational advantages of our algorithm over the prior ones.

[1]Salesforce Research [2] B-DAT Lab and CICAEET, Nanjing University of Information Science & Technology, Nanjing, 210044, China. Correspondence to: Xiao-Tong Yuan <xtyuan@nuist.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

1. Introduction

We consider the following ℓ_2-regularized empirical risk minimization (ERM) problem:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} F(\boldsymbol{\theta}) := \frac{1}{n}\sum\nolimits_{i=1}^{n} \ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i) + \frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2, \quad (1)$$

where $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ is a training set; the convex loss function $\ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i)$ measures the discrepancy between the linear prediction $\boldsymbol{\theta}^\top \boldsymbol{x}_i$ and the ground truth $\boldsymbol{y}_i$; and the regularization term $\frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2$ aims at enhancing generalization ability of the linear model. In the field of statistical learning, the formulation (1) encapsulates a vast body of problems including least squares regression, logistic regression and softmax regression, to name a few. In this work, we focus on developing scalable and autonomous first-order optimization methods to solve this fundamental problem, which has been extensively studied with a bunch of efficient algorithms proposed including gradient descent (GD) (Cauchy, 1847), stochastic GD (SGD) (Robbins & Monro, 1951), SDCA (Shalev-Shwartz, 2012), SVRG (Johnson & Zhang, 2013), Catalyst (Lin et al., 2015), SCSG (Lei & Jordan, 2017) and Katyusha (Allen-Zhu, 2017).

Motivation. Despite the remarkable success of the stochastic gradient methods and their variance-reduced extensions, the stochastic gradient evaluation complexity (which usually dominates the computational cost) of these algorithms tends to scale linearly with data size n. Such a linear dependence is not only expensive when data scale is huge but also problematic in online and life-long learning regimes where samples are coming infinitely. As pointed out in (Lei & Jordan, 2017), there are situations in which accurate solutions can be obtained with less than a single pass through the data, *e.g.* for a large-scale dataset with similar and redundant samples. Therefore, developing data-size-independent learning algorithms is of special importance in big data era.

Particularly, we are interested in efficiently optimizing problem (1) to its intrinsic excess error bound which typically scales as $\mathcal{O}(1/\sqrt{n})$. As shown in (Bottou & Bousquet, 2008), the excess error, which measures the expected prediction discrepancy between the optimum model and the learnt model over all possible samples and thus reflects the generalization performance of the model, can be decomposed into model approximation error, estimation error and

Table 1: Comparison of IFO complexity for first-order stochastic algorithms on the μ-strongly-convex problem (1) with condition number κ. The solution $\boldsymbol{\theta}$ with ϵ-optimization-error is measured by sub-optimality $\mathbb{E}\left[F(\boldsymbol{\theta})-F(\boldsymbol{\theta}^*)\right]\leq\epsilon$ with optimum $F(\boldsymbol{\theta}^*)$. Here we define a set of constants for quadratic (generic) loss: $\beta_1=1.5$ (2.25), $\beta_2=1$ (2), $\beta_3=3$ (4.5), $\beta_4=1$ (2.5), $\beta_5=1$ (1.5), $\gamma=1.5$ (2.25). These different constants only affects the logarithm factor $\xi=\log\left(\frac{1}{\epsilon}\right)$. For brevity, we define $\Theta=\frac{\kappa^{1.5}\xi^{\gamma}}{\epsilon^{0.25}}+\frac{1}{\epsilon}$. The third column summarizes the conditions under which HSDMPG has lower IFO complexity than the compared algorithms.

	ϵ-Optimization Error for ERM (1)		$\frac{1}{\sqrt{n}}$-Optimization Error for ERM (1)
	IFO Complexity	Better Zoom of HSDMPG	
SGD	$\mathcal{O}\left(\frac{1}{\mu\epsilon}\right)$	① $\mu\leq 1 \& \mu\kappa^{1.5}\epsilon^{0.75}\xi^{\beta_1}\leq\mathcal{O}(1)$ or ② $\mathcal{O}(n)\leq\frac{1}{\mu\epsilon\xi^{\beta_2}}\wedge\frac{1}{\kappa^2\mu^2\epsilon^2\xi^{\beta_3}}$	$\mathcal{O}(n)$
SVRG, SAGA, APSDCA	$\mathcal{O}\left((n+\kappa)\log\left(\frac{1}{\epsilon}\right)\right)$	① $\Theta\xi^{-1}\leq\mathcal{O}(n)$	$\mathcal{O}(n\log(n))$
APCG	$\mathcal{O}\left(\frac{n}{\sqrt{\mu}}\log\left(\frac{1}{\epsilon}\right)\right)$	① $\Theta\mu^{0.5}\xi^{-1}\leq\mathcal{O}(n)$ or ② $\mu^{-1}\kappa^2\xi^{\beta_4}\leq\mathcal{O}(n)$	$\mathcal{O}\left(n^{1.25}\log(n)\right)$
SPDC, Catalyst, Katyusha	$\mathcal{O}\left((n+\sqrt{n\kappa})\log\left(\frac{1}{\epsilon}\right)\right)$	① $\Theta\xi^{-1}\wedge\Theta^2\xi^{-2}\kappa^{-1}\leq\mathcal{O}(n)$	$\mathcal{O}(n\log(n))$
AMSVRG	$\mathcal{O}\left(\left(n+\frac{n\kappa}{n+\sqrt{\kappa}}\right)\log\left(\frac{1}{\epsilon}\right)\right)$	① $\Theta\xi^{-1}\leq\mathcal{O}(n)$	$\mathcal{O}(n\log(n))$
Varag	$\mathcal{O}\left(n\log\left(n\wedge\frac{1}{\epsilon}\right)+\sqrt{n}\left(\frac{1}{\epsilon^{0.5}}\wedge\kappa^{0.5}\log\left(\frac{1}{\epsilon\kappa}\right)\right)\right)$	① $\Theta\log^{-1}(n\wedge\frac{1}{\epsilon})\leq\mathcal{O}(n)$ or ② $\Theta^2(\epsilon\vee\frac{1}{\kappa\log^2(\frac{1}{\epsilon\kappa})})\leq\mathcal{O}(n)$	$\mathcal{O}(n\log(n))$
SCSG	$\mathcal{O}\left((n\wedge\frac{\kappa}{\epsilon}+\kappa)\log\left(\frac{1}{\epsilon}\right)\right)$	① $\Theta\xi^{-1}\leq\mathcal{O}(n)\leq\frac{\kappa}{\epsilon}$ or ② $\kappa\epsilon^{1.5}\xi^{\beta_5}\leq\mathcal{O}(1)\&\frac{\kappa}{\epsilon}\leq\mathcal{O}(n)$	$\mathcal{O}(n\log(n))$
HSDMPG quadratic	$\mathcal{O}\left(\frac{\kappa^{1.5}\epsilon^{0.75}\log^{1.5}(\frac{1}{\epsilon})+1}{\epsilon}\wedge\left(\kappa\sqrt{n}\log^{1.5}\left(\frac{1}{\epsilon}\right)+n\log\left(\frac{1}{\epsilon}\right)\right)\right)$	———	$\mathcal{O}\left(n^{0.875}\log^{1.5}(n)\right)$
HSDMPG generic	$\mathcal{O}\left(\frac{\kappa^{1.5}\epsilon^{0.75}\log^{2.25}(\frac{1}{\epsilon})+1}{\epsilon}\wedge\left(\kappa\sqrt{n}\log^{2.5}\left(\frac{1}{\epsilon}\right)+n\log^2\left(\frac{1}{\epsilon}\right)\right)\right)$	———	$\mathcal{O}\left(n^{0.875}\log^{2.25}(n)\right)$

optimization error. Among them, the model approximation error measures how closely the selected predication model can approximate the optimal model; the estimation error measures the prediction effects of minimizing the empirical risk instead of the population risk; the optimization error denotes the prediction difference between the exact and approximate solutions of ERM. Therefore, to achieve small excess error, one should minimize the three terms jointly. With optimal choice $\mu=\mathcal{O}(1/\sqrt{n})$ to balance empirical risk and generalization gap, the estimation error is known to be at the order of $\mathcal{O}(1/\sqrt{n})$, which implies the excess error is dominated by $\mathcal{O}(1/\sqrt{n})$ (Vapnik, 2006; Shalev-Shwartz et al., 2009; Shalev-Shwartz & Ben-David, 2014). Thus, it is sufficient to optimize the regularized ERM problem (1) to the optimization error $\mathcal{O}(1/\sqrt{n})$ to match the optimal excess error without redundant computation.

Overiew of our contribution. The main contribution of this paper is a novel Hybrid Stochastic-Deterministic Minibatch Proximal Gradient (HSDMPG) algorithm with substantially improved data-size-independent complexity over existing methods. For quadratic problems, the core idea of our method is to recurrently convert the original large-scale ERM problem into a series of minibatch proximal ERM subproblems for efficient minimization and update. Specifically, as a starting point, we uniformly randomly select a minibach $\mathcal{S}$ of components of the risk function F to form a stochastic approximation $F_{\mathcal{S}}$ that will be fixed throughout the algorithm iteration. Next, at each iteration step, we first construct a stochastic surrogate of F by combining the Bregman divergence of $F_{\mathcal{S}}$ at the current iterate and a first-order hybrid stochastic-deterministic approximation of F; and then we invoke existing variance-reduced algorithms, such as SVRG, to minimize this surrogate subproblem to desired optimization error. For quadratic loss, we can provably establish sharper bounds of incremental first order oracle (IFO, see Definition 2) for such a hybrid stochastic-deterministic minibatch proximal update procedure in large-scale settings. To extend the strong efficiency guarantee to generic strongly convex losses, we propose to iteratively convert the non-quadratic problem into a sequence of quadratic subproblems such that the aforementioned method can be readily applied for optimization. In this way, up to logarithmic factors, HSDMPG still enjoys an identical sharp bound of IFO for strongly convex problems.

Table 1 summarizes the computational complexity (measured by IFO) of HSDMPG and several representative baselines, including SGD (Robbins & Monro, 1951; Shamir, 2011), SVRG (Johnson & Zhang, 2013), SAGA (Defazio et al., 2014), APSDCA (Shalev-Shwartz & Zhang, 2014), APCG (Lin et al., 2014), SPDC (Zhang & Xiao, 2015), Catalyst (Lin et al., 2015), Varag (Lan et al., 2019), AMSVRG (A. Nitanda, 2016), Katyusha (Allen-Zhu, 2017),

SCSG (Lei & Jordan, 2017). In the following, we highlight the advantages of our method over these prior approaches:

- To achieve ϵ-optimization-error, *i.e.* $\mathbb{E}[F(\boldsymbol{\theta})–F(\boldsymbol{\theta}^*)] \leq \epsilon$, the IFO complexity of HSDMPG on problem (1) is $\mathcal{O}\Big(\frac{\kappa^{1.5}\epsilon^{0.75}\log^{\tau_1}(\frac{1}{\epsilon})+1}{\epsilon} \wedge \Big(\kappa\sqrt{n}\log^{\tau_2}\big(\frac{1}{\epsilon}\big)+n\log^{\tau_3}\big(\frac{1}{\epsilon}\big)\Big)\Big)$ where $\tau_1 = 1.5$, $\tau_2 = 1.5$ and $\tau_3 = 1$ for quadratic loss and $\tau_1 = 2.25$, $\tau_2 = 2.5$ and $\tau_3 = 2$ for generic strongly convex loss. In comparison, the IFO complexity bounds of all the compared algorithms except SGD and SCSG scale linearly w.r.t. the data size n. As specified in the third column of Table 1, HSDMPG is superior to these algorithms in large-scale problem settings which are of central interest in big data applications. Compared with SGD, since in most cases, the condition number κ is at the order of $\mathcal{O}(1/\mu)$, HSDMPG improves over SGD by a factor at least $\mathcal{O}\big(\kappa \wedge \frac{1}{\kappa^{0.5}\epsilon^{0.75}}\big)$ (up to logarithm factors). For SCSG, HSDMPG also shows higher computational efficiency when (1) the optimization error ϵ is small which corresponds to conditions ① or ② in Table 1; and (2) the data size n is large which corresponds to condition ③ in Table 1.
- For the practical setting where $\epsilon = \mathcal{O}(1/\sqrt{n})$ which matches the optimal intrinsic excess error, HSDMPG has the IFO complexity $\mathcal{O}\left(n^{0.875}\log^{1.5}(n)\right)$ for the quadratic loss and $\mathcal{O}\left(n^{0.875}\log^{2.25}(n)\right)$ for the generic strongly convex loss. By ignoring the small logarithm term $\log(n)$, both complexities of HSDMPG are lower than the complexity bound $\mathcal{O}\big(n\big)$ of SGD by a factor $\mathcal{O}\big(n^{0.125}\big)$. Similarly, HSDMPG respectively improves over APCG and other remaining algorithms, such as SVRG, Katyusha, Varag and SCSG, by factors of $\mathcal{O}\big(n^{0.375}\big)$ and $\mathcal{O}\big(n^{0.125}\big)$. These results demonstrate the superior computational efficiency of HSDMPG for attaining near-optimal generalization rate of a statistical learning model.

2. Related Work

Stochastic gradient algorithms. Gradient descent (GD) (Cauchy, 1847) method has long been applied to solve ERM and enjoys linear convergence rate on strongly convex problems. But it needs to compute full gradient per iteration, leading to huge computation cost on large-scale problems. To improve efficiency, incremental gradient algorithms have been developed via leveraging the finite-sum structure and have witnessed tremendous progress recently. For instance, SGD (Robbins & Monro, 1951; Bottou, 1991) only evaluates gradient of one (or a minibatch) randomly selected sample at each iteration, which greatly reduces the cost of each iteration and shows more appealing efficiency than GD on large-scale problems (Shamir, 2011; A. Nitanda, 2016; Hendrikx et al., 2019; Mohammadi et al., 2019). Along this line of research, a variety of variance-reduced variants, such as SVRG (Johnson & Zhang, 2013), SAGA (Defazio et al., 2014), APSDCA (Shamir, 2011), AMSVRG (A. Nitanda, 2016), SCSG (Lei & Jordan, 2017), Catalyst (Lin et al., 2015), Katyusha (Allen-Zhu, 2017), Varag (Lan et al., 2019), are developed and have delivered exciting progress such as linear convergence rates on strongly convex problems as opposed to sublinear rates of vanilla SGD (Shamir, 2011). The hybrid stochastic-deterministic gradient descent method (Friedlander & Schmidt, 2012; Zhou et al., 2018a;b; Mokhtari et al., 2016; Mokhtari & Ribeiro, 2017) iteratively samples an evolving minibatch of samples for gradient estimation or subproblem construction and works favorably in reducing the computational complexity. Our HSDMPG method differs significantly from these prior algorithms. Based on the Bregman-divergence of the minibatch function and a hybrid stochastic-deterministic first-order approximation of the original function, HSDMPG constructs a variance-reduced minibatch proximal function which is provably more efficient. Moreover, HSDMPG can employ any off-the-shelf algorithms to solve the constructed sub-problems in the inner loop and thus is flexible for implementation. HSDMPG shares a similar spirit with the DANE method (Shamir et al., 2014) which also uses a local Bregman-divergence-based function approximation for communication-efficient distributed quadratic loss optimization. The main difference lies in the way of constructing first-order approximation of the risk function: HSDMPG employs a novel hybrid stochastic-deterministic approximation strategy which is substantially more efficient than the deterministic strategy as used by DANE.

Generalization and optimization. In the seminal work of Bottou & Bousquet (2008), it has been demonstrated that the excess error that measures the generalization performance of an ERM model over a function class can be decomposed into three terms in expectation: an *approximation error* that measures how accurate the function class can approximate the underlying optimum model; an *estimation error* that measures the effects of minimizing ERM instead of population risk; and an *optimization error* that represents the difference between the exact solution and the approximate solution of ERM. Particularly, for the ℓ_2-regularized convex ERM with linear models as in (1), its estimation error (or excess risk) has long been studied with a vast body of deep theoretical results established (Shalev-Shwartz & Ben-David, 2014; Hardt et al., 2016; Bach & Moulines, 2013; Dieuleveut et al., 2017; Zhou & Feng, 2018a;b). A simple yet powerful tool for analyzing estimation error is the *stability* of an estimator to the changes of training dataset (Bousquet & Elisseeff, 2002). The ℓ_2-regularized convex ERM has been shown to have uniform stability of order $\mathcal{O}(1/(\mu n))$ (Bousquet & Elisseeff, 2002), which then gives rise to the optimal choice $\mu = \mathcal{O}(1/\sqrt{n})$

to balance empirical loss and generalization gap to achieve estimation error $\mathcal{O}(1/\sqrt{n})$ (Shalev-Shwartz et al., 2009; Feldman & Vondrak, 2019). This implies that the overall excess error is dominated by $\mathcal{O}(1/\sqrt{n})$. In this sense, it suffices to solve the ℓ_2-regularized ERM to optimization error $\mathcal{O}(1/\sqrt{n})$ to match the intrinsic excess error.

3. Hybrid Stochastic-Deterministic Minibatch Proximal Gradient

In this section, we first introduce the hybrid stochastic-deterministic minibatch proximal gradient (HSDMPG) algorithm for quadratic loss function along with convergence rate and computational complexity analysis. Then, we extend HSDMPG and its theoretical analysis to generic strongly convex loss functions.

3.1. The HSDMPG method for quadratic loss

3.1.1. Algorithm

The HSDMPG method is outlined in Algorithm 1. The initial step is to randomly sample a minibatch $\mathcal{S}$ of data points of size s to construct a stochastic approximation

$$F_{\mathcal{S}}(\boldsymbol{\theta}) = \frac{1}{s}\sum\nolimits_{i\in\mathcal{S}} \ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i) + \frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2 \qquad (2)$$

to the original risk function $F(\boldsymbol{\theta})$ in problem (1). $F_{\mathcal{S}}(\boldsymbol{\theta})$ will be fixed throughout the computational procedure to follow. Then in the iteration loop the algorithm iterates between two steps of S1 and S2. In step S1, we uniformly randomly sample a size increasing minibatch $\mathcal{S}_t$ of samples to estimate an inexact function $F_{\mathcal{S}_t}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}_t|}\sum_{i\in\mathcal{S}_t} \ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i) + \frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2$. Let $\mathcal{D}_g(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = g(\boldsymbol{\theta}_1) - g(\boldsymbol{\theta}_2) - \langle \nabla g(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\rangle$ denote the Bregman divergence of a function g. Based on $F_{\mathcal{S}}(\boldsymbol{\theta})$ and $F_{\mathcal{S}_t}(\boldsymbol{\theta})$, we construct a variance-reduced minibatch proximal objective $\widetilde{P}_{t-1}(\boldsymbol{\theta})$ to approximate the objective $F(\boldsymbol{\theta})$ in (1), where $\widetilde{P}_{t-1}(\boldsymbol{\theta}) \triangleq$

$$F_{\mathcal{S}_t}(\boldsymbol{\theta}_{t-1}) + \langle \nabla F_{\mathcal{S}_t}(\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}\rangle + \mathcal{D}_{\widetilde{F}_{\mathcal{S}}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{t-1}).$$

Here $\mathcal{D}_{\widetilde{F}_{\mathcal{S}}}(\boldsymbol{\theta}, \boldsymbol{\theta}_{t-1})$ is the Bregman divergence of a regularized loss $\widetilde{F}_{\mathcal{S}}(\boldsymbol{\theta}) = F_{\mathcal{S}}(\boldsymbol{\theta}) + \frac{\gamma}{2}\|\boldsymbol{\theta}\|_2^2$ which essentially measures the distance between $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{t-1}$ on the current geometry curve estimated on $\widetilde{F}_{\mathcal{S}}(\boldsymbol{\theta})$. We define the next iterate as

$$\boldsymbol{\theta}_t = \arg\min\nolimits_{\boldsymbol{\theta}} \widetilde{P}_{t-1}(\boldsymbol{\theta}) = \arg\min\nolimits_{\boldsymbol{\theta}} P_{t-1}(\boldsymbol{\theta}), \qquad (3)$$

where $P_{t-1}(\boldsymbol{\theta}) \triangleq$

$$F_{\mathcal{S}}(\boldsymbol{\theta}) + \langle \nabla F_{\mathcal{S}_t}(\boldsymbol{\theta}_{t-1}) - \nabla F_{\mathcal{S}}(\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta}\rangle + \frac{\gamma}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}\|_2^2.$$

In P_{t-1}, its finite-sum structure comes from the initial stochastic approximation $F_{\mathcal{S}}(\boldsymbol{\theta})$ and its gradient at $\boldsymbol{\theta}_{t-1}$.

Algorithm 1 Hybrid Stochastic-Deterministic Minibatch Proximal Gradient (HSDMPG) for quadratic loss.

Input: initialization $\boldsymbol{\theta}_0$, regularization constant γ in (3), optimization error ε_t.
Initialization: Uniformly randomly sample a data batch $\mathcal{S}$ of size s to form $F_{\mathcal{S}}(\boldsymbol{\theta})$ in (2).
for $t = 1, 2, \ldots, T$ **do**
(S1) Uniformly randomly sample a minibatch $\mathcal{S}_t$ to form $F_{\mathcal{S}_t}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}_t|}\sum_{i\in\mathcal{S}_t} \ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i) + \frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2$ and compute $\nabla F_{\mathcal{S}_t}(\boldsymbol{\theta}_{t-1})$ to construct loss $P_{t-1}(\boldsymbol{\theta})$ in (3).
(S2) Optimize the subproblem (3), *e.g.* via SVRG, to obtain $\boldsymbol{\theta}_t$ that satisfies $\|\nabla P_{t-1}(\boldsymbol{\theta}_t)\|_2 \le \varepsilon_t$.
end for
Output: $\boldsymbol{\theta}_T$.

Since along with more iterations, the size of $\mathcal{S}_t$ increases which indicates that the loss P_{t-1} is a variance-reduced loss and will converge to the original loss $F(\boldsymbol{\theta})$ in problem (1). Then in step S2, we approximately solve problem (3) via a stochastic gradient optimization method such as SVRG. The principle behind this strategy is that for the initial optimization progress, inexact gradient already can well decrease the loss since the current solution is far from the optimum, while along more iterations, the current solution becomes closer to optimum, requiring more accurate gradient for further reducing the loss function. In this way, our proposed method can well balance the converge speed and the computational cost at each iteration and thus has the potential to achieve improved overall computational efficiency. Shamir et al. (2014) has proposed the DANE method which uses a similar local Bregman divergence based regularization for distributed quadratic optimization problems. Our method improves upon DANE in two aspects: 1) we use variance-reduction techniques to reduce the overall computational complexity, and 2) HSDMPG is applicable not only to quadratic problems but also to generic strongly convex problems with about the same computational complexity as discussed in Sec. 3.2.

3.1.2. Convergence and complexity analysis

We first introduce two necessary definitions, namely strong convexity and Lipschitz smoothness, which are conventionally used in the analysis of convex optimization methods (Shamir, 2011; Johnson & Zhang, 2013).

Definition 1 (Strong Convexity and Smoothness)**.** *A differentiable function $g(\boldsymbol{\theta})$ is said to be μ-strongly-convex and L-smooth if $\forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2$, it satisfies*

$$\frac{\mu}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2 \le \mathcal{D}_g(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \le \frac{L}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2.$$

where $\mathcal{D}_g(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = g(\boldsymbol{\theta}_1) - g(\boldsymbol{\theta}_2) - \langle \nabla g(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\rangle$.

For brevity, let $\boldsymbol{H}$ be the Hessian matrix of the quadratic function $F(\boldsymbol{\theta})$ and $\ell_i(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i) + \frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2$. Denote

$\|\boldsymbol{\theta}\|_{\boldsymbol{H}} = \sqrt{\boldsymbol{\theta}^\top \boldsymbol{H}\boldsymbol{\theta}}$. In the analysis to follow, we always suppose that $\|\boldsymbol{x}_i\| \leq r, \forall i$, which generally holds for natural data analysis, *e.g.*, in computer vision and signal processing. We summarize our main result in Theorem 1 which shows the linear convergence rate of HSDMPG for quadratic problems. See proof in Appendix B.1.

Theorem 1. *Assume each loss $\ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i)$ is quadratic and L-smooth w.r.t. $\boldsymbol{\theta}^\top \boldsymbol{x}_i$, and $\sup_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^n \|\boldsymbol{H}^{-1/2}(\nabla F(\boldsymbol{\theta}) - \nabla \ell_i(\boldsymbol{\theta}))\|_2^2 \leq \nu^2$. By setting $\gamma = (\sqrt{\log(d)} + \sqrt{2})Lr^2/\sqrt{s}$, $\varepsilon_t = \frac{\mu^{1.5}}{4(\mu+2\gamma)} \exp\big(-\frac{\mu(t-1)}{2(\mu+2\gamma)}\big)$, $|\mathcal{S}_t| = \frac{16\nu^2(\mu+2\gamma)^2}{\mu^2}\exp\big(\frac{\mu t}{2(\mu+2\gamma)}\big) \wedge n$, where d is the problem dimension, the sequence $\{\boldsymbol{\theta}_t\}$ produced by Algorithm 1 satisfies*

$$\mathbb{E}[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)] = \tfrac{1}{2}\mathbb{E}[\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_{\boldsymbol{H}}^2] \leq \zeta \exp\big(-\tfrac{\mu t}{\mu+2\gamma}\big),$$

where $\zeta = \frac{1}{2}\left(\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\|_{\boldsymbol{H}} + \frac{1}{2}\right)^2 + \frac{5}{8}$.

The main message conveyed by Theorem 1 is that HSDMPG enjoys linear convergence rate on the quadratic loss when we use evolving size of the minibatch $\mathcal{S}_t$. Note here we only assume each loss $\ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i)$ is L-smooth w.r.t. $\boldsymbol{\theta}^\top \boldsymbol{x}_i$. This assumption is much milder than the smoothness assumption on the function $F(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ which is used in other algorithm analysis, such as SGD and SVRG. The assumption that $\sup_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^n \|\boldsymbol{H}^{-1/2}(\nabla F(\boldsymbol{\theta}) - \nabla \ell_i(\boldsymbol{\theta}))\|_2^2 \leq \nu^2$ in HSDMPG is mild, which requires the variance of stochastic gradient under the Hessian matrix is bounded. Such an assumption is analogous to the one used in analysis of SGD that imposing the bounded-variance assumption on stochastic gradient, namely, $\frac{1}{n}\sum_{i=1}^n \|\nabla F(\boldsymbol{\theta}) - \nabla \ell_i(\boldsymbol{\theta})\|_2^2$.

Based on this result, we further analyze the computational complexity of HSDMPG to better understand its overall efficiency in computation. At each iteration, we use the SVRG method solve the inner-loop subproblem (3) because it only accesses the first-order information of the objective function and is efficient. Following (Johnson & Zhang, 2013; Zhang & Xiao, 2015; Zhou et al., 2019; Shen et al., 2019), we employ the incremental first order oracle (IFO) complexity as the computation complexity metric for solving the finite-sum solving problem (1).

Definition 2. *An IFO takes an index $i \in [n]$ and a point $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, and returns the pair $(\ell_i(\boldsymbol{\theta}), \nabla \ell_i(\boldsymbol{\theta}))$.*

The IFO complexity can accurately reflect the overall computational performance of a first-order algorithm, as objective value and gradient evaluation usually dominate the per-iteration complexity. Based on these preliminaries, we summarize our main result on the computation complexity of HSDMPG in Corollary 1 with proof provided in Appendix B.2.

Corollary 1 (Computation complexity of HSDMPG for quadratic loss)**.** *Suppose that the assumptions in Theorem 1 hold and the inner-loop subproblems are solved via SVRG, then the IFO complexity of* HSDMPG *on the quadratic loss to achieve $\mathbb{E}[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)] \leq \epsilon$ is of the order $\mathcal{O}\Big(\big(1 + \frac{\kappa^3 \log^{1.5}(d)}{s^{1.5}}\big)\frac{\nu^2}{\epsilon} \bigwedge \big(1 + \frac{\kappa \log^{0.5}(d)}{s^{0.5}}\big) n \log\big(\frac{1}{\epsilon}\big) + \kappa\sqrt{s\log(d)}\log^2\big(\frac{1}{\epsilon}\big)\Big)$, where $\kappa = L/\mu$ denotes the conditional number.*

According to Corollary 1, by choosing s as $s = \frac{\kappa\nu\log^{0.5}(d)}{\epsilon^{0.5}\log(1/\epsilon)} \wedge n$ or $s = \frac{n}{\log(1/\epsilon)}$ and ignoring the constant ν and the logarithm factor $\log(d)$ of the problem dimension d, the IFO complexity of HSDMPG is at the order of

$$\mathcal{O}\Big(\frac{\kappa^{1.5}\epsilon^{0.75}\log^{1.5}(\frac{1}{\epsilon}) + 1}{\epsilon} \wedge \Big(\kappa\sqrt{n}\log^{1.5}\big(\frac{1}{\epsilon}\big) + n\log\big(\frac{1}{\epsilon}\big)\Big)\Big).$$

One may compare such a complexity with the state-of-the-arts listed in Table 1. Compared with those algorithms in the table whose IFO complexity scales linearly with the data size n, *e.g.* SVRG, APCG, Katyusha and AMSVRG, the proposed HSDMPG has data-size-independent IFO complexity and can outperform them for large-scale learning problems where the data size n could be huge. To be more precise, the third column of Table 1 summarizes the conditions under which HSDMPG outperforms these algorithms in terms of computational complexity. For the algorithms whose IFO complexity does not depend on n, namely SGD and SCSG, HSDMPG also enjoys substantially lower complexity in most cases. Concretely, since κ is typically at the order of $\mathcal{O}(1/\mu)$, when $\kappa \leq \epsilon^{1.5}$ which holds for moderately larger κ, HSDMPG improves over SGD by a factor at least $\mathcal{O}\big(\kappa \wedge \frac{1}{\kappa^{0.5}\epsilon^{0.75}}\big)$ (up to the logarithmic factor). As for SCSG, HSDMPG also achieves higher efficiency when (1) the optimization error is small which corresponds to conditions ① in the third column of Table 1, (2) the sampler size n is large which corresponds to condition ②. These results show that HSDMPG is well suited for solving large-scale learning problems.

From the perspective of generalization, we are particularly interested in the computational complexity of HSDMPG for optimizing the ℓ_2-ERM model (1) to its intrinsic excess error bound which characterizes the generalization performance of the model. As reviewed in Section 2, the excess error of the considered ℓ_2-ERM model is typically of order $\mathcal{O}(1/\sqrt{n})$. Accordingly, one only needs to solve the optimization problem to the optimization error $\epsilon = \mathcal{O}(1/\sqrt{n})$ (Bottou & Bousquet, 2008; Shalev-Shwartz et al., 2009). Moreover, to accord with this intrinsic excess error bound, the regularization constant μ should also be at the order of $\mathcal{O}(\frac{1}{\sqrt{n}})$. In this way, the condition number κ could scale as large as $\mathcal{O}(\sqrt{n})$. Based on these results and Corollary 1, we can derive the IFO complexity bound of HSDMPG for this case in Corollary 2.

Corollary 2. *Suppose that the assumptions in Corol-*

Algorithm 2 Hybrid Stochastic-Deterministic Minibatch Proximal Gradient (HSDMPG) on the generic loss.

Input: Regularization constant γ and initialization $\boldsymbol{\theta}_0$.
for $t = 1, 2, \ldots, T$ **do**
(S1) Construct a finite-sum quadratic function $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta})$ in Eqn. (4) to approximate $F(\boldsymbol{\theta})$ at $\boldsymbol{\theta}_{t-1}$.
(S2) Run Algorithm 1 with regularization constant γ and initialization $\boldsymbol{\theta}_{t-1}$ to minimize the finite-sum function $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta})$ such that $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta}_t) \leq \min_{\boldsymbol{\theta}} \boldsymbol{Q}_{t-1}(\boldsymbol{\theta}) + \varepsilon'_t$.
end for
Output: $\boldsymbol{\theta}_T$.

lary 1 hold. By setting $s = \mathcal{O}\big(\frac{\nu n^{0.75}\log^{0.5}(d)}{\log(n)}\big)$, *the IFO complexity of* HSDMPG *on the quadratic loss to achieve* $\mathbb{E}[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)] \leq \frac{1}{\sqrt{n}}$ *is at the order of* $\mathcal{O}\big(\nu^{0.5} n^{0.875}\log^{0.75}(d)\log^{1.5}(n) + \nu^2 n^{0.5}\big)$.

See its proof in Appendix B.3. From Corollary 2, one can observe that the IFO complexity of HSDMPG for quadratic problems is at the order of $\mathcal{O}\left(n^{0.875}\log^{1.5}(n)\right)$. It means that HSDMPG can reach the intrinsic excess error $\mathcal{O}\big(1/\sqrt{n}\big)$ with strictly less than a single pass over the entire training dataset. In comparison, we can observe from Table 1 that in the same practical setting, SGD and APCG have IFO complexity $\mathcal{O}(n)$ and $\mathcal{O}\left(n^{1.25}\log(n)\right)$ respectively. By ignoring the logarithm factor $\log(n)$ which is much smaller than n for large-scale learning problems, HSDMPG improves over these two methods by factors $\mathcal{O}\big(n^{0.125}\big)$ and $\mathcal{O}\big(n^{0.375}\big)$, respectively. The IFO complexity of all other algorithms in Table 1, including SVRG, SCSG, SPDC, APSDCA, AMSVRG, Catalyst, Katyusha and Varag, are all at the order of $\mathcal{O}(n\log(n))$. Similarly, by ignoring the logarithmic factors, HSDMPG has lower IFO complexity than these algorithms by a factor $\mathcal{O}\big(n^{0.125}\big)$. To summarize this group of results comparison, HSDMPG would be significantly superior to all these state-of-the-arts when solving quadratic optimization problems to intrinsic excess error.

3.2. Algorithm for generic convex loss function

The computational complexity guarantees established in the previous section are only applicable to quadratic loss function. In order to extend these results to non-quadratic convex loss function, we apply a quadratic approximation strategy to convert the original non-quadratic problem into a sequence of quadratic optimization sub-problems such that each of the subproblem can be optimized by HSDMPG. More specifically, suppose that the loss function $\ell(\boldsymbol{\theta}^\top\boldsymbol{x}, \boldsymbol{y})$ is twice differentiable w.r.t. $\boldsymbol{\theta}^\top\boldsymbol{x}$ and is L-smooth w.r.t. $\boldsymbol{\theta}^\top\boldsymbol{x}$. Then we can verify that $\nabla^2 F(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n \ell''(\boldsymbol{\theta}^\top\boldsymbol{x}_i, \boldsymbol{y}_i)\boldsymbol{x}_i\boldsymbol{x}_i^\top + \mu\boldsymbol{I} \preceq \bar{\boldsymbol{H}} \triangleq \frac{L}{n}\sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i^\top + \mu\boldsymbol{I}$ for all $\boldsymbol{\theta}$. Therefore, at each iteration, we construct an upper bound of the second-order Taylor expansion of F at $\boldsymbol{\theta}_{t-1}$ as expressed by $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta}) \triangleq$

$$F(\boldsymbol{\theta}_{t-1}) + \langle \nabla F(\boldsymbol{\theta}_{t-1}), \boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}\rangle + \Delta_{t-1}(\boldsymbol{\theta}), \quad (4)$$

where $\Delta_{t-1}(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})^\top \bar{\boldsymbol{H}}(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})$. The finite-sum structure in $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta})$ comes from $\nabla F(\boldsymbol{\theta}_{t-1}) = \frac{1}{n}\sum_{i=1}^n \nabla\ell(\boldsymbol{\theta}^\top\boldsymbol{x}_i, \boldsymbol{y}_i) + \mu\boldsymbol{\theta}$ and $\bar{\boldsymbol{H}}$. Thus we can estimate $\boldsymbol{\theta}_t$ by applying HSDMPG to the quadratic function $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta})$ with a warm-start initialization $\boldsymbol{\theta}_{t-1}$ such that

$$\boldsymbol{Q}_{t-1}(\boldsymbol{\theta}_t) \leq \min_{\boldsymbol{\theta}} \boldsymbol{Q}_{t-1}(\boldsymbol{\theta}) + \varepsilon'_t. \quad (5)$$

The above nested-loop computation procedure is summarized in Algorithm 2. We remark that when computing the gradient of $\boldsymbol{Q}_{t-1}(\boldsymbol{\theta})$, we can compute the gradient associated with $\bar{\boldsymbol{H}}$ at the point $\boldsymbol{\theta}$ as $\bar{\boldsymbol{H}}(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}) = \frac{L}{n}\sum_{i=1}^n(\boldsymbol{x}_i^\top(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}))\boldsymbol{x}_i + \mu(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})$ which only computes the inner-product $\boldsymbol{x}_i^\top(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})$ without explicitly computing $\bar{\boldsymbol{H}}$. In this way, the computational cost of each stochastic gradient associated with $\bar{\boldsymbol{H}}$ is actually much cheaper than that of computing stochastic gradient of $\nabla F(\boldsymbol{\theta}_{t-1})$, since the former only involves vector products and the later one is usually complicated, *e.g.* involving the exponential computation in logistic regression. Then we establish Theorem 2 to guarantee the convergence of Algorithm 2 and analyze its computational complexity. See Appendix C.1 for a proof of this main result.

Theorem 2 (Convergence rate and computation complexity of HSDMPG for generic loss)**.** *Suppose that each loss function* $\ell(\boldsymbol{\theta}^\top\boldsymbol{x}, \boldsymbol{y})$ *is* L*-smooth and* σ*-strongly convex w.r.t.* $\boldsymbol{\theta}^\top\boldsymbol{x}$. *By setting* $\varepsilon'_t = \frac{\sigma}{2L}\exp\left(-\frac{\sigma}{2L}t\right)$, *the sequence* $\{\boldsymbol{\theta}_t\}$ *produced by Algorithm 2 satisfies*

$$F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*) \leq \exp\Big(-\frac{\sigma t}{2L}\Big)\big(1 + F(\boldsymbol{\theta}_0) - F(\boldsymbol{\theta}^*)\big).$$

Suppose the assumptions in Corollary 1 hold. Then by setting $\kappa = \frac{L}{\mu}$ *the IFO complexity of Algorithm 2 to achieve* $\mathbb{E}\left[F(\boldsymbol{\theta}_t) - F(\boldsymbol{\theta}^*)\right] \leq \epsilon$ *is at the order of* $\mathcal{O}\Big(\Big(1 + \frac{\kappa^3\log^{1.5}(d)}{s^{1.5}}\Big)\frac{L\nu^2}{\sigma\epsilon} \bigwedge \Big(1 + \frac{\kappa\log^{0.5}(d)}{s^{0.5}}\Big)\frac{L^3 n}{\sigma^3}\log^2\big(\frac{1}{\epsilon}\big) + \frac{L^2\sqrt{s\log(d)}}{\sigma\mu}\log^3\big(\frac{1}{\epsilon}\big)\Big)$.

Theorem 2 suggests that the objective $F(\boldsymbol{\theta}_t)$ converges linearly to the optimum $F(\boldsymbol{\theta}^*)$ with rate $\exp(-\frac{\sigma}{2L})$. Note that σ is the strong convexity parameter of the loss function $\ell(\boldsymbol{\theta}^\top\boldsymbol{x}, \boldsymbol{y})$ w.r.t. $\boldsymbol{\theta}^\top\boldsymbol{x}$ instead of $\boldsymbol{\theta}$ which is usually not relying on data scale for widely used loss functions such as the logistic loss (Yuan & Li, 2019) and thus leads to fast outer-loop convergence rate. In contrast, the strong convexity parameter μ of the risk function F is typically set at the order of $\mathcal{O}\big(1/\sqrt{n}\big)$ so as to match the intrinsic excess error.

In terms of computational complexity, by choosing the proper value of s from $s = \frac{\nu\kappa\log^{0.5}(d)}{\epsilon^{0.5}\log^{1.5}(1/\epsilon)} \wedge n$ and $s =$

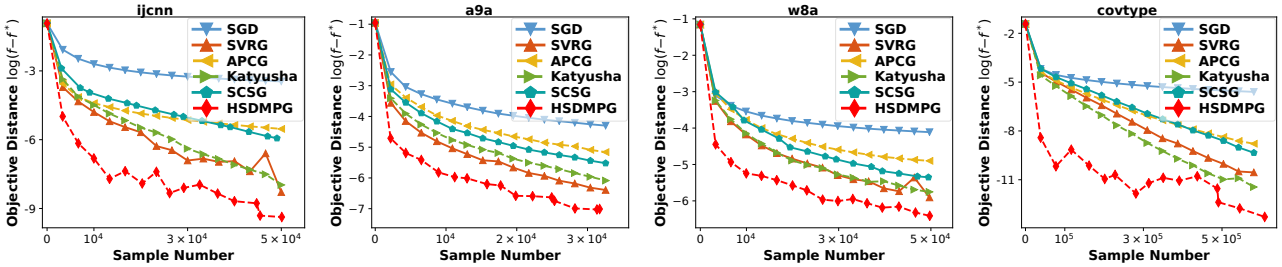

Figure 1: Single-epoch processing: stochastic gradient algorithms process data a single pass on quadratic problems.

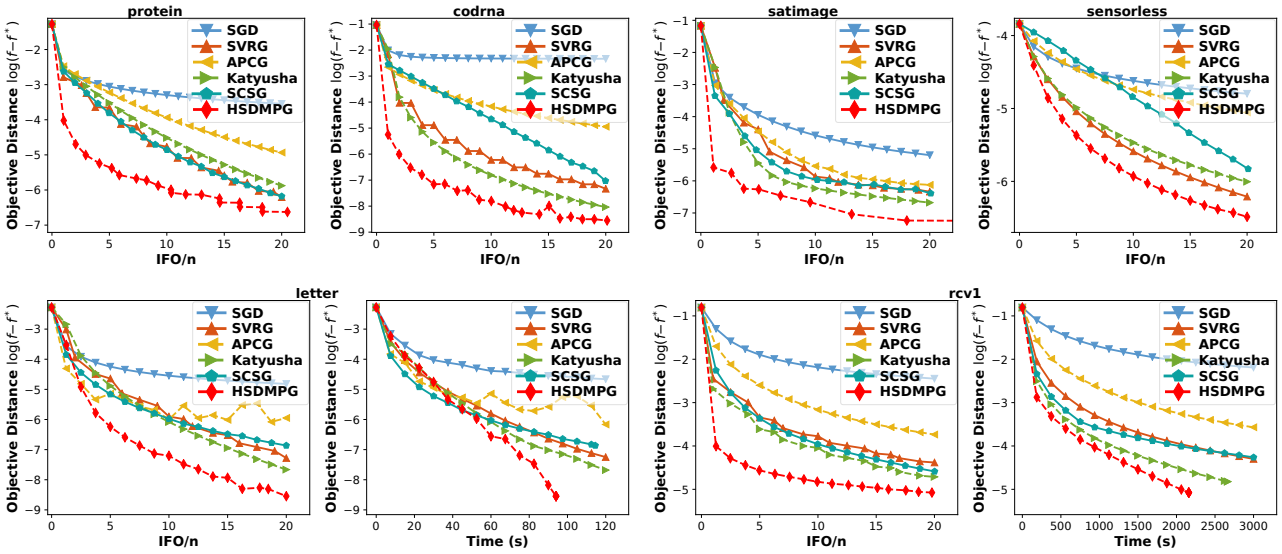

Figure 2: Multi-epoch processing: stochastic gradient algorithms process data multiple pass on quadratic problems.

$\frac{\mu L\kappa n}{\sigma^2\log(1/\epsilon)} \wedge n$ in Algorithm 1, the IFO complexity of HSDMPG for generic convex loss can be shown to scale as

$$\mathcal{O}\Big(\frac{\kappa^{1.5}\epsilon^{0.75}\log^{2.25}(\frac{1}{\epsilon})+1}{\epsilon}\wedge\Big(\kappa\sqrt{n}\log^{2.5}(\frac{1}{\epsilon})+n\log^2(\frac{1}{\epsilon})\Big)\Big)$$

Compared with the methods listed in Table 1, one can observe that for generic strongly convex problems, HSDMPG enjoys lower computational complexity than all the compared algorithms except SGD and SCSG for large-scale learning problems where the sample number n is sufficiently large to satisfy the conditions in the third column of Table 1. Similar to the results on quadratic loss, HSDMPG improves over SGD by a factor at least $\mathcal{O}\big(\kappa\wedge\frac{1}{\kappa^{0.5}\epsilon^{0.75}}\big)$. So when the optimization error ϵ is very small or the condition number κ is large, HSDMPG will be much more efficient than SGD. For SCSG, HSDMPG is of higher efficiency in two regimes, namely (1) the optimization error is small which corresponds to conditions ① or ② in Table 1, and (2) the sampler number n is large which corresponds to condition ③. These results show the advantages HSDMPG in solving large-scale strongly-convex learning problems.

Finally we consider a realistic case where the optimization error of problem (1) matches the intrinsic excess error bound $\mathcal{O}(1/\sqrt{n})$. For this case, as discussed at the end of Section 3.1.2 that the regularization parameter should be set at the scale of $\mu=\mathcal{O}(1/\sqrt{n})$ with balanced impact against the guarantees on estimation error. As a result, the condition number κ could scale as large as $\mathcal{O}(\sqrt{n})$. The following corollary substantializes the IFO complexity bound in Theorem 2 to such a setting. See Appendix C.2 for a proof of this result.

Corollary 3. *Suppose the assumptions in Theorem 2 hold. By setting* $s=\mathcal{O}\big(\frac{\nu n^{0.75}\log^{0.5}(d)}{\log(n)}\big)$, *the IFO complexity of* HSDMPG *on the generic loss to achieve* $\mathbb{E}[F(\boldsymbol{\theta}_t)-F(\boldsymbol{\theta}^*)]\leq\frac{1}{\sqrt{n}}$ *is of order* $\mathcal{O}\big(\nu^{0.5}n^{0.875}\log^{0.75}(d)\log^{2.25}(n)+\nu^2n^{0.5}\big)$.

Corollary 3 shows that for generic convex loss, the IFO complexity of HSDMPG to attain the $\mathcal{O}\big(1/\sqrt{n}\big)$ intrinsic excess error is of the order $\mathcal{O}\left(n^{0.875}\log^{2.25}(n)\right)$. This shows that HSDMPG is able to achieve nearly optimal generalization with less than a single pass over data. Compared with the complexity bound for the quadratic loss, such a more general IFO complexity bound of HSDMPG only comes at the cost of a slightly increased overhead on the logarithmic factor, *i.e.*, from $\log^{1.5}(n)$ for the quadratic case to the $\log^{2.25}(n)$ for generic convex loss. Similar to the ob-

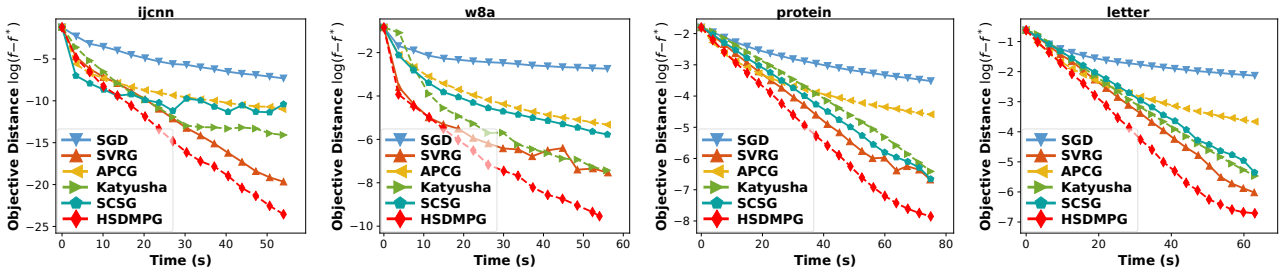

Figure 3: Multi-epoch processing (about 8 epochs): stochastic gradient algorithms process data multiple pass on logistic regression problems (ijcnn and w08) and softmax regression problems (protein and letter).

servations in the quadratic case, from results in Table 1 one can observe that all the considered state-of-the-art methods need to process the entire data at least one pass to achieve the desired optimization error for generic convex loss. All in all, the established theoretical results for both quadratic and non-quadratic loss functions showcase the benefit of HSDMPG for efficient optimization of large-scale learning problems with near-optimal generalization.

4. Experiments

In this section, we carry out experiments to compare the numerical performance of HSDMPG with several representative stochastic gradient optimization algorithms, including SGD (Robbins & Monro, 1951), SVRG (Johnson & Zhang, 2013), APCG (Lin et al., 2014), Katyusha (Allen-Zhu, 2017) and SCSG (Lei & Jordan, 2017). We evaluate all the considered algorithms on two sets of strongly-convex learning tasks. The first set is for ridge regression with least squared loss $\ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i) = \frac{1}{2}\|\boldsymbol{\theta}^\top \boldsymbol{x}_i - \boldsymbol{y}_i\|_2^2$, where $\boldsymbol{y}_i$ is the target output of sample $\boldsymbol{x}_i$. In the second setting we consider two classification models: logistic regression with loss $\ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i) = \log\left(1+\exp(-\boldsymbol{y}_i \boldsymbol{\theta}^\top \boldsymbol{x}_i)\right)$ and multi-class softmax regression with k-classification loss $\ell(\boldsymbol{\theta}^\top \boldsymbol{x}_i, \boldsymbol{y}_i) = \sum_{j=1}^k \mathbf{1}\{\boldsymbol{y}_i = j\} \log\left(\frac{\exp(\boldsymbol{\theta}_j^\top \boldsymbol{x}_i)}{\sum_{s=1}^k \exp(\boldsymbol{\theta}_s^\top \boldsymbol{x}_i)}\right)$. We run simulations on ten datasets whose details are described in Appendix D.4. For HSDMPG, we set the size s of $\mathcal{S}$ around $n^{0.75}$. For the minibatch for inner problems, we set initial minibatch size $|\mathcal{S}_1| = 50$ and then follow our theory to exponentially expand size of $\mathcal{S}_t$ with proper exponential rate. The regularization constant in the subproblem (3) is set to be $\gamma = \sqrt{\log(d)/s}$ as suggested by our theory. The optimization error ε_t in (3) is controlled by respectively allowing SVRG to run 3 epochs and 10 epochs on the two sets of tasks. Similarly, we control the optimization error ε_t' in (5) by running SVRG with 3 epochs. Since there is no ground truth on real data, we run FGD sufficiently long until $\|\nabla F(\tilde{\boldsymbol{\theta}})\|_2 \leq 10^{-10}$ and take $F(\tilde{\boldsymbol{\theta}})$ as an approximate optimal value $F(\boldsymbol{\theta}^*)$ for sub-optimality estimation.

4.1. Results for the quadratic loss

Single-epoch evaluation results. Here we first evaluate well-conditioned quadratic problems such that moderately accurate solution can be obtained after only one epoch of data pass. Such a one epoch setting usually occurs in online learning. Towards this goal, we set the regularization parameter $\mu = 0.01$ to make the quadratic problems well-conditioned. From Figure 1, one can observe that HSDMPG exhibits much sharper convergence behavior than the considered baselines, though most algorithms can achieve small optimization error after one epoch processing of data. This confirms the theoretical predictions in Corollaries 1 and 2 that HSDMPG is cheaper in IFO complexity than SGD and variance-reduced algorithms, *e.g.* SVRG and SCSG, when the data scale is large.

Multi-epoch evaluation results. For more challenging problems, an algorithm usually requires multiple cycles of data processing to achieve accurate optimization. Here we reset the regularization strength parameter in quadratic problems as $\mu = 10^{-4}$ for generating more challenging optimization tasks. As shown in Figure 2, one can again observe that HSDMPG converges faster than all the compared algorithms in terms of IFO complexity. Particularly, we compare both IFO complexity and wall-clock running time on the letter and rcv11 datasets. The convergence curves under these two metrics consistently show the superior computational efficiency of HSDMPG to the considered state-of-the-arts on large-scale learning tasks, which well support the theoretical predictions in Corollaries 1 and 2.

4.2. Results for the non-quadratic loss

Finally, we investigate the convergence performance of the proposed HSDMPG on non-quadratic convex loss functions. Specifically, we evaluate all the compared algorithms on logistic regression and its multi-classes version, *i.e.* softmax regression, in which their regularization modulus parameters are set as $\mu = 0.01$. Figure 3 reports the running time evolving curves which can accurately reflects the efficiency of an algorithm. These results show that HSDMPG converges significantly faster than the baseline

algorithms for the considered non-quadratic loss functions, which well support the predictions in Theorem 2 and Corollary 3 that HSDMPG has lower IFO complexity than the state-of-the-arts in the regimes where data scale is large. This set of results also demonstrates the effectiveness of our sequential quadratic-approximation approach for extending the attractive computational complexity guarantees on quadratic loss to generic convex loss.

5. Conclusions

We proposed HSDMPG as a hybrid stochastic-deterministic minibach proximal gradient method for ℓ_2-regularized ERM problems. For quadratic loss, we showed that HSDMPG enjoys provably lower computational complexity than prior state-of-the-art SVRG algorithms in large-scale settings. Particularly, to attain the optimization error $\epsilon = \mathcal{O}\big(1/\sqrt{n}\big)$ at the order of intrinsic excess error bound of ERM which is sufficient for generalization, the stochastic gradient complexity of HSDMPG is dominated by $\mathcal{O}(n^{0.875})$ (up to logarithmic factors). To our best knowledge, HSDMPG for the first time achieves nearly optimal generalization in less than a single pass over data. Almost identical computational complexity guarantees hold for an extension of HSDMPG to generic strongly convex loss functions via sequential quadratic approximation. Extensive numerical results demonstrate the substantially improved computational efficiency of HSDMPG over the prior methods. We expect that the algorithms and computational learning theory developed in this paper for ℓ_2-regularized ERM can be extended to stochastic convex optimization problems. Also, it is worthwhile to explore the opportunity of using first-order acceleration techniques to further improve the computational complexity guarantees of HSDMPG.

Acknowledgements

The authors sincerely thank the anonymous reviewers for their constructive comments on this work. Xiao-Tong Yuan is supported in part by National Major Project of China for New Generation of AI under Grant No.2018AAA0100400 and in part by Natural Science Foundation of China (NSFC) under Grant No.61876090 and No.61936005.

References

A. Nitanda, A. Accelerated stochastic gradient descent for minimizing finite sums. In *Artificial Intelligence and Statistics*, pp. 195–203, 2016.

Allen-Zhu, Z. Katyusha: The First Direct Acceleration of Stochastic Gradient Methods. In *ACM SIGACT Symposium on Theory of Computing*, 2017.

Bach, F. and Moulines, E. Non-strongly-convex smooth stochastic approximation with convergence rate o (1/n). In *Proc. Conf. Neural Information Processing Systems*, pp. 773–781, 2013.

Bottou, L. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12, 1991.

Bottou, L. and Bousquet, O. The tradeoffs of large scale learning. In *Proc. Conf. Neural Information Processing Systems*, pp. 161–168, 2008.

Bousquet, O. and Elisseeff, A. Stability and generalization. *J. of Machine Learning Research*, 2(Mar):499–526, 2002.

Cauchy, M. A. Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptesrendus des séances de l'Académie des sciences de Paris*, 25:536–538, 1847.

Defazio, A., Bach, F., and Lacoste-Julien, S. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Proc. Conf. Neural Information Processing Systems*, pp. 1646–1654, 2014.

Dieuleveut, A., Flammarion, N., and Bach, F. Harder, better, faster, stronger convergence rates for least-squares regression. *J. of Machine Learning Research*, 18(1):3520–3570, 2017.

Feldman, V. and Vondrak, J. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In *Conf. on Learning Theory*, pp. 1270–1279, 2019.

Friedlander, M. P. and Schmidt, M. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In *Proc. Int'l Conf. Machine Learning*, 2016.

Hendrikx, H., Bach, F., and Massoulié, L. Asynchronous accelerated proximal stochastic gradient for strongly convex distributed finite sums. *arXiv preprint arXiv:1901.09865*, 2019.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Proc. Conf. Neural Information Processing Systems*, pp. 315–323, 2013.

Lan, G., Li, Z., and Zhou, Y. A unified variance-reduced accelerated gradient method for convex optimization. In *Proc. Conf. Neural Information Processing Systems*, pp. 10462–10472, 2019.

Lei, L. and Jordan, M. Less than a single pass: Stochastically controlled stochastic gradient. In *Artificial Intelligence and Statistics*, pp. 148–156, 2017.

Lin, H., Mairal, J., and Harchaoui, Z. A universal catalyst for first-order optimization. In *Proc. Conf. Neural Information Processing Systems*, pp. 3384–3392, 2015.

Lin, Q., Lu, Z., and Xiao, L. An accelerated proximal coordinate gradient method. In *Proc. Conf. Neural Information Processing Systems*, pp. 3059–3067, 2014.

Mohammadi, H., Razaviyayn, M., and Jovanović, M. Robustness of accelerated first-order algorithms for strongly convex optimization problems. *arXiv preprint arXiv:1905.11011*, 2019.

Mokhtari, A. and Ribeiro, A. First-order adaptive sample size methods to reduce complexity of empirical risk minimization. In *Proc. Conf. Neural Information Processing Systems*, pp. 2060–2068, 2017.

Mokhtari, A., Daneshmand, H., Lucchi, A., Hofmann, T., and Ribeiro, A. Adaptive newton method for empirical risk minimization to statistical accuracy. In *Proc. Conf. Neural Information Processing Systems*, pp. 4062–4070, 2016.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3): 400–407, 1951.

Shalev-Shwartz, S. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Shalev-Shwartz, S. and Zhang, T. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *Proc. Int'l Conf. Machine Learning*, pp. 64–72, 2014.

Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. Stochastic convex optimization. In *Conf. on Learning Theory*, 2009.

Shamir, O. Making gradient descent optimal for strongly convex stochastic optimization. *CoRR abs/1109.5647*, 2011.

Shamir, O., Srebro, N., and Zhang, T. Communication-efficient distributed optimization using an approximate newton-type method. In *Proc. Int'l Conf. Machine Learning*, pp. 1000–1008, 2014.

Shen, Z., Zhou, P., Fang, C., and Ribeiro, A. A stochastic trust region method for non-convex minimization. *arXiv preprint arXiv:1903.01540*, 2019.

Vapnik, V. *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

Yuan, X. and Li, P. On convergence of distributed approximate newton methods: Globalization, sharper bounds and beyond. *arXiv preprint arXiv:1908.02246*, 2019.

Zhang, Y. and Xiao, L. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *Proc. Int'l Conf. Machine Learning*, pp. 353–361, 2015.

Zhou, P. and Feng, J. Understanding generalization and optimization performance of deep cnns. In *Proc. Int'l Conf. Machine Learning*, 2018a.

Zhou, P. and Feng, J. Empirical risk landscape analysis for understanding deep neural networks. In *Int'l Conf. Learning Representations*, 2018b.

Zhou, P., Yuan, X., and Feng, J. Efficient stochastic gradient hard thresholding. In *Proc. Conf. Neural Information Processing Systems*, 2018a.

Zhou, P., Yuan, X., and Feng, J. New insight into hybrid stochastic gradient descent: Beyond with-replacement sampling and convexity. In *Proc. Conf. Neural Information Processing Systems*, pp. 1234–1243, 2018b.

Zhou, P., Yuan, X., and Feng, J. Faster first-order methods for stochastic non-convex optimization on riemannian manifolds. In *Int'l Conf. Artificial Intelligence and Statistics*, 2019.