Optimal Stochastic Strongly Convex Optimization with a Logarithmic Number of Projections

Jianhui Chen[1], Tianbao Yang[2], Qihang Lin[3], Lijun Zhang[4], and Yi Chang[1]

[1]Yahoo Research, Sunnyvale, CA 94089, USA
[2]Department of Computer Science, The University of Iowa, Iowa City, IA 52242, USA
[3]Department of Management Sciences, The University of Iowa, Iowa City, IA 52242, USA
[4]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

Abstract

We consider stochastic strongly convex optimization with a complex inequality constraint. This complex inequality constraint may lead to computationally expensive projections in algorithmic iterations of the stochastic gradient descent (SGD) methods. To reduce the computation costs pertaining to the projections, we propose an Epoch-Projection Stochastic Gradient Descent (Epro-SGD) method. The proposed Epro-SGD method consists of a sequence of epochs; it applies SGD to an augmented objective function at each iteration within the epoch, and then performs a projection at the end of each epoch. Given a strongly convex optimization and for a total number of T iterations, Epro-SGD requires only $\log(T)$ projections, and meanwhile attains an optimal convergence rate of $O(1/T)$, both in expectation and with a high probability. To exploit the structure of the optimization problem, we propose a proximal variant of Epro-SGD, namely Epro-ORDA, based on the optimal regularized dual averaging method. We apply the proposed methods on real-world applications; the empirical results demonstrate the effectiveness of our methods.

1 INTRODUCTION

Recent years have witnessed an increased interest in adopting the stochastic (sub)gradient (SGD) methods [1, 3, 21] for solving large-scale machine learning problems. In each of the algorithmic iterations, SGD reduces the computation cost by sampling one (or a small number of) example for computing a stochastic (sub)gradient. Thus the computation cost in SGD is independent of the size of the data available for training; this property makes SGD appealing for large-scale optimization. However, when the optimization problems involve a complex domain (for example a positive definite constraint or a polyhedron one), the projection operation in each iteration of SGD, which is used to ensure the feasibility of the intermediate solutions, may become the computational bottleneck.

In this paper we consider to solve the following constrained optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^d} \quad & f(\mathbf{x}) \\ s.t. \quad & c(\mathbf{x}) \leq 0, \end{aligned} \tag{1}$$

where $f(\mathbf{x})$ is β-strongly convex [23] and $c(\mathbf{x})$ is convex. We assume a stochastic access model for $f(\cdot)$, in which the only access to $f(\cdot)$ is via a stochastic gradient oracle; in other words, given arbitrary $\mathbf{x}$, this stochastic gradient oracle produces a random vector $\mathbf{g}(\mathbf{x})$, whose expectation is a subgradient of $f(\cdot)$ at the point $\mathbf{x}$, i.e., $\mathrm{E}[\mathbf{g}(\mathbf{x})] \in \partial f(\mathbf{x})$, where $\partial f(\mathbf{x})$ denotes the subdifferential set of $f(\cdot)$ at $\mathbf{x}$. On the other hand we have the full access to the (sub)gradient of $c(\cdot)$.

The standard SGD method [5] solves Eq. (1) by iterating the updates in Eq. (2) with an appropriate step size η_t, e.g., $\eta_t = 1/(\beta t)$), as below

$$\mathbf{x}_{t+1} = \mathcal{P}_{\{\mathbf{x}\in\mathbb{R}^d: c(\mathbf{x})\leq 0\}} \left[\mathbf{x}_t - \eta_t \mathbf{g}(\mathbf{x}_t)\right], \tag{2}$$

and then returning $\widehat{\mathbf{x}}_T = \sum_{t=1}^T \mathbf{x}_t/T$ as the final solution for a total number of iterations T. Note that $\mathcal{P}_{\mathcal{D}}[\widehat{\mathbf{x}}]$ is a projection operator defined as

$$\mathcal{P}_{\mathcal{D}}[\widehat{\mathbf{x}}] = \arg\min_{\mathbf{x}\in\mathcal{D}} \|\mathbf{x} - \widehat{\mathbf{x}}\|_2^2. \tag{3}$$

If the involved constraint function $c(\mathbf{x})$ is complex (e.g., a polyhedral or a positive definite constraint), computing the associated projection may be computationally expensive; for example, a projection onto a positive definite cone over $\mathbb{R}^{d\times d}$ requires a full singular value decomposition (SVD) operation with time complexity of $O(d^3)$.

In this paper, we propose an epoch-based SGD method, called Epro-SGD, which requires only a logarithmic number of projections (onto the feasible set), and meanwhile achieves an optimal convergence rate for stochastic strongly convex optimization. Specifically, the proposed Epro-SGD method consists of a sequence of epochs; within each of the epochs, the standard SGD is applied to optimize a composite objective function augmented by the complex constraint function, hence avoiding the expensive projections steps; at the end of every epoch, a projection operation is performed to ensure the feasibility of the intermediate solution. Our analysis shows that given a strongly convex optimization and for a total number of T iterations, Epro-SGD requires only $\log(T)$ projections, and meanwhile achieves an optimal rate of convergence at $O(1/T)$, both in expectation and with a high probability.

To exploit the structure (for example the sparisty) of the optimization problem, we propose a proximal variant of the Epro-SGD method, namely Epro-ORDA, which utilizes an existing optimal dual averaging method to solve the involved proximal mapping. Our analysis shows that Epro-ORDA similarly requires only a logarithmic number of projections while enjoys an optimal rate of convergence.

For illustration we apply the proposed Epro-SGD methods on two real-world applications, i.e., the constrained Lasso formulation and the large margin nearest neighbor (LMNN) classification. Our experimental results demonstrate the efficiency of the proposed methods, in comparison to the existing methods.

2 RELATED WORK

The present work is inspired from the break-through work in [20], which proposed two novel one-projection-based stochastic gradient descent (OneProj) methods for stochastic convex optimizations. Specifically the first OneProj method was developed for general convex optimization; it introduces a regularized Lagrangian function as

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda c(\mathbf{x}) - \frac{\gamma}{2}\lambda^2, \quad \lambda \geq 0,$$

then applies SGD to the convex-concave problem $\min_{\mathbf{x}\in\mathcal{B}} \max_{\lambda\geq 0} L(\mathbf{x}, \lambda)$, and finally performs only one projection at the end of all iterations, where $\mathcal{B}$ is a bounded ball subsuming $\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^d : c(\mathbf{x}) \leq 0\}$ as a subset.

The second OneProj method was developed for strongly convex optimization. The proposed method introduced an augmented objective function

$$F(\mathbf{x}) = f(\mathbf{x}) + \gamma \ln\left(1 + \exp\left(\frac{\lambda c(\mathbf{x})}{\gamma}\right)\right), \qquad (4)$$

where γ is a parameter dependent on the total number of iterations T, and λ is a problem specific parameter [20]. OneProj applies SGD to the augmented objective function, specifically using a stochastic subgradient of $f(\mathbf{x})$ and a subgradient of $c(\mathbf{x})$, and then performs a projection step after all iterations. For a total number T iterations, the OneProj method achieves a rate of convergence at $O(\log T/(\beta T))$, which is suboptimal for stochastic strongly convex optimization.

Several recent works [15, 26] propose optimal methods with optimal rates of convergence at $O(1/T)$ for stochastic strongly convex optimization. In particular, the Epoch-SGD method [15] consists of a sequence of epochs, each of which has a geometrically decreasing step size and a geometrically increasing iteration number. This method however needs to project the intermediate solutions onto a feasible set at every algorithmic iteration; when the involved constraint is complex, the involved projection is usually computationally expensive. This limitation restricts the practical applications on large scale data analysis. Therefore we are motivated to develop an optimal stochastic algorithm for strongly convex optimization but with a constant number of projections.

Another closely related work is the logT-SGD [33] for stochastic strongly convex and smooth optimization. LogT-SGD achieves an optimal rate of convergence, while requires to perform $O(\kappa \log_2 T)$ projections, where κ is the ratio of the smoothness parameter to the strong convexity parameter. There are several key differences between our proposed Epro-SGD method and logT-SGD: (i) logT-SGD and its analysis rely on both the smoothness and the strong convexity of the objective function; in contrast, Epro-SGD only assumes that the objective function is strongly convex; (ii) the number of the required projections in logT-SGD is $O(\kappa \log_2 T)$, where the conditional number κ can be very large in real applications; in contrast, Epro-SGD requires at most $\log_2 T$ projections.

Besides reducing the number of projections in SGD, another line of research is based on the conditional gradient algorithms [7, 14, 17, 18, 32]; this type of algorithms mostly build upon the Frank-Wolfe technique [11], which eschews the projection in favor of a linear optimization step; however in general, they require the smoothness assumption in the objective function. On the other hand, [12, 16] extended Frank-Wolfe techniques to stochastic or online setting for general and strongly convex optimizations. Specifically [16] presents an online/stochastic Frank-Wolfe (OFW) algorithm with a convergence rate $O(1/T^{1/3})$ for general convex optimization problems, which is slower than the optimal rate $O(1/\sqrt{T})$. [12] presents an algorithm for online strongly convex optimization with an $O(\log T)$ regret bound, implying an $O(\log T/T)$ convergence rate for stochastic stronlgy convex optimization. This algorithm requires the problem domain to be a polytope, instead of a convex inequality constraint used in this paper; it also hinges on an efficient local linear optimization oracle that amounts to approximately solving a linear optimization problem over an intersection of a ball and and the feasible domain; furthermore the convergence result only holds in expectation and is sub-optimal.

3 EPOCH-PROJECTION SGD ALGORITHM

In this section, we present an epoch-projection SGD method, called Epro-SGD, for solving Eq. (1) and discuss its convergence result. Based on a stochastic dual averaging algorithm, we then present a proximal variant of the proposed Epro-SGD method.

3.1 SETUP AND BACKGROUND

Denote the optimal solution to Eq. (1) by $\mathbf{x}_*$ and its domain set by $\mathcal{D} = \{\mathbf{x} \in \mathbb{R}^d : c(\mathbf{x}) \leq 0\}$. Since $f(\mathbf{x})$ is β-strongly convex [23] and $c(\mathbf{x})$ is convex, the optimization problem in Eq. (1) is strongly convex. Note that the strong convexity in $f(\cdot)$ implies that $f(\mathbf{x}) \geq f(\mathbf{x}_*) + (\beta/2)\|\mathbf{x} - \mathbf{x}_*\|^2$ for any $\mathbf{x}$. Our analysis is based on the following assumptions:

A1. The stochastic subgradient $\mathbf{g}(\mathbf{x})$ is uniformly bounded by G_1, i.e., $\|\mathbf{g}(\mathbf{x})\|_2 \leq G_1$.

A2. The subgradient $\partial c(\mathbf{x})$ is uniformly bounded by G_2, i.e., $\|\partial c(\mathbf{x})\|_2 \leq G_2$ for any $\mathbf{x}$.

A3. There exists a positive value $\rho > 0$ such that

$$\left[\min_{c(\mathbf{x})=0,\, \mathbf{v}\in\partial c(\mathbf{x}),\, \mathbf{v}\neq 0} \|\mathbf{v}\|_2\right] \geq \rho. \qquad (5)$$

Remarks Assumptions A1 and A2 respectively impose an upper bound on the stochastic subgradient of the objective function $f(\cdot)$ and the constraint function $c(\cdot)$. Assumption A3 ensures that the projection of a point onto a feasible domain does not deviate too much from this intermediate point. Note that Assumption A1 is previously used in [15]; a condition similar to Assumption A3 is used in [20], which however simply assumes that $\min_{c(\mathbf{x})=0} \|\nabla c(\mathbf{x})\|_2 \geq \rho$, without considering possible non-differentiability in $c(\cdot)$.

A key consequence of Assumption A3 is presented in the following lemma.

Lemma 1. *For any $\widehat{\mathbf{x}}$, let $\widetilde{\mathbf{x}} = \arg\min_{c(\mathbf{x})\leq 0} \|\mathbf{x} - \widehat{\mathbf{x}}\|_2^2$. If Assumption A3 holds, then*

$$\|\widehat{\mathbf{x}} - \widetilde{\mathbf{x}}\|_2 \leq \frac{1}{\rho}[c(\widehat{\mathbf{x}})]_+, \quad \rho > 0, \qquad (6)$$

where $[s]_+$ is a hinge operator defined as $[s]_+ = s$ if $s \geq 0$, and $[s]_+ = 0$ otherwise.

Proof. If $c(\widehat{\mathbf{x}}_T) \leq 0$, we have $\widehat{\mathbf{x}} = \widetilde{\mathbf{x}}$; the inequality in Eq. (6) trivially holds. If $c(\widehat{\mathbf{x}}_T) > 0$, we can verify that $c(\widetilde{\mathbf{x}}_T) = 0$, and there exists $s \geq 0$ and $\mathbf{v} \in \partial c(\widetilde{\mathbf{x}}_T)$ such that $\widetilde{\mathbf{x}}_T - \widehat{\mathbf{x}}_T + s\mathbf{v} = 0$ (using duality theory). It follows that $\widehat{\mathbf{x}}_T - \widetilde{\mathbf{x}}_T = s\mathbf{v}$ ($\mathbf{v} \neq 0$), and thus $\widehat{\mathbf{x}}_T - \widetilde{\mathbf{x}}_T$ is the same direction as $\mathbf{v}$. It follows that

$$\begin{aligned} c(\widehat{\mathbf{x}}_T) &= c(\widehat{\mathbf{x}}_T) - c(\widetilde{\mathbf{x}}_T) \geq (\widehat{\mathbf{x}}_T - \widetilde{\mathbf{x}}_T)^\top \mathbf{v} \\ &= \|\mathbf{v}\|_2 \|\widehat{\mathbf{x}}_T - \widetilde{\mathbf{x}}_T\|_2 \geq \rho \|\widehat{\mathbf{x}}_T - \widetilde{\mathbf{x}}_T\|_2, \end{aligned}$$

where the last inequality uses Assumption A3. This completes the proof of this lemma. □

The result in Lemma 1 is closely related to *the polyhedral error bound condition* [13, 31]; this condition shows that the distance of a point to the optimal set of a convex optimization problem whose epigraph is a polyhedron is bounded by the distance of the objective value at this point to the optimal objective value scaled by a constant. For illustration, we consider the optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} [c(\mathbf{x})]_+$$

with an optimal set as $\{\mathbf{x} \in \mathbb{R}^d : c(\mathbf{x}) \leq 0\}$. If $c(\widehat{\mathbf{x}}) > 0$, $\widetilde{\mathbf{x}} = \arg\min_{c(\mathbf{x})\leq 0} \|\mathbf{x} - \widehat{\mathbf{x}}\|_2^2$ is the closest point in the optimal set to $\widehat{\mathbf{x}}$. Therefore, by *the polyhedral error bound condition* of a polyhedral convex optimization, if $c(\mathbf{x})$ is a polyhedral function, there exists a $\rho > 0$ such that

$$\|\widehat{\mathbf{x}} - \widetilde{\mathbf{x}}\|_2 \leq \frac{1}{\rho}\left([c(\widehat{\mathbf{x}})]_+ - \min_{\mathbf{x}}[c(\mathbf{x})]_+\right) = \frac{1}{\rho}[c(\widehat{\mathbf{x}})]_+.$$

Below we present three examples in which Assumption A3 or Lemma 1 is satisfied. Example 1: an affine function $c(\mathbf{x}) = \mathbf{c}^\top\mathbf{x} - b$ with $\rho = \|\mathbf{c}\|_2$. Example 2: the ℓ_1 norm constraint $\|\mathbf{x}\|_1 \leq B$ where $\rho = \min_{\mathbf{x}:\|\mathbf{x}\|_1 = B} \|\partial\|\mathbf{x}\|_1\|_2 \geq 1$. Example 3: the maximum of a finite number of affine functions $c(\mathbf{x}) = \max_{1\leq i\leq m} \mathbf{c}_i^\top\mathbf{x} - b_i$ satisfying Lemma 1 as well as the polyhedral error bound condition [31].

3.2 MAIN ALGORITHM

To solve Eq. (1) (using Epro-SGD), we introduce an augmented objective function by incorporating the constraint function as

$$F(\mathbf{x}) = f(\mathbf{x}) + \lambda[c(\mathbf{x})]_+. \qquad (7)$$

It is worth noting that the augmented function in Eq. (7) does not have any iteration-dependent parameter, for example the parameter γ in Eq. (4). λ is a prescribed parameter satisfying $\lambda > G_1/\rho$, as illustrated in Lemma 2.

The details of our proposed Epro-SGD algorithm is presented in Algorithm 1. Similar to Epoch-SGD [15], Epro-SGD consists of a sequence of epochs, each of which has a geometrically decreasing step size and a geometrically increasing iteration number (Line 9 in Algorithm 1). The updates in every intra-epoch (Line 5 - 6) are standard SGD steps applied to the augmented objective function $F(\mathbf{x})$ with $\mathbf{x} = \mathbf{x}_t^k$. Epro-SGD is different from Epoch-SGD in that the former computes a projection only at the end of each epoch, while the latter computes a projection at each iteration. Consequently, when the projection step is computationally expensive (e.g., projecting onto a positive definite constraint), Epro-SGD may require much less computation time than Epoch-SGD.

Algorithm 1 Epoch-projection SGD (Epro-SGD)

```
Input: an initial step size $\eta_1$, total number of iterations $T$, and number of iterations in the first epoch $T_1$, a Lagrangian multiplier $\lambda$ ($\lambda > G_1/\rho$)
Initialization: $\mathbf{x}_1^1 \in \mathcal{D}$ and $k = 1$
while $\sum_{i=1}^{k} T_i \leq T$ do
    for $t = 1, \ldots, T_k$ do
        Compute a stochastic gradient $\mathbf{g}(\mathbf{x}_t^k)$
        Compute $\mathbf{x}_{t+1}^k = \mathbf{x}_t^k - \eta_k(\mathbf{g}(\mathbf{x}_t^k) + \lambda\partial[c(\mathbf{x}_t^k)]_+)$
    end for
    Compute $\widetilde{\mathbf{x}}_T^k = \mathcal{P}_{\mathcal{D}}[\widehat{\mathbf{x}}_T^k]$, where $\widehat{\mathbf{x}}_T^k = \sum_{t=1}^{T_k} \mathbf{x}_t^k / T_k$
    Update $\mathbf{x}_1^{k+1} = \widetilde{\mathbf{x}}_T^k, T_{k+1} = 2T_k, \eta_{k+1} = \eta_k/2$
    Set $k = k + 1$
end while
```

In Lemma 2, we present an important convergence analysis for the intra-epoch steps of Algorithm 1, which are key building blocks for deriving the main results in Theorem 1.

Lemma 2. *Under Assumptions A1$\sim$A3, if we apply the update $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta(\widetilde{\nabla} f(\mathbf{x}_t; \varepsilon_t) + \lambda\nabla[c(\mathbf{x}_t)]_+)$ for a number of T iterations, the following equality holds*

$$\mathrm{E}[f(\widetilde{\mathbf{x}}_T)] - f(\mathbf{x}_*) \leq \mu \left[\eta(G_1^2 + \lambda^2 G_2^2) + \frac{\mathrm{E}[\|\mathbf{x}_1 - \mathbf{x}_*\|_2^2]}{2\eta T}\right],$$

where $\mu = \rho/(\rho - G_1/\lambda)$.

Proof. Let $F(\mathbf{x}) = f(\mathbf{x}) + \lambda[c(\mathbf{x})]_+$ and denote by $\mathrm{E}_t[X]$ the expectation conditioned on the randomness until round $t-1$. It is easy to verify that $F(\mathbf{x}) \geq f(\mathbf{x})$, $F(\mathbf{x}) \geq f(\mathbf{x}) + \lambda c(\mathbf{x})$ and $F(\mathbf{x}_*) = f(\mathbf{x}_*)$. For any $\mathbf{x}$, we have

$$\begin{aligned}(\mathbf{x}_t - \mathbf{x})^\top \nabla F(\mathbf{x}_t) \leq\ & \frac{1}{2\eta}\left(\|\mathbf{x} - \mathbf{x}_t\|_2^2 - \|\mathbf{x} - \mathbf{x}_{t+1}\|_2^2\right) + \\ & \frac{\eta}{2}\|\widetilde{\nabla} f(\mathbf{x}_t, \xi_t) + \lambda\nabla[c(\mathbf{x}_t)]_+\|_2^2 + \\ & (\mathbf{x} - \mathbf{x}_t)^\top(\mathbf{g}(\mathbf{x}_t) - \nabla f(\mathbf{x}_t)) \\ \leq\ & \frac{1}{2\eta}\left(\|\mathbf{x} - \mathbf{x}_t\|_2^2 - \|\mathbf{x} - \mathbf{x}_{t+1}\|_2^2\right) + \\ & \eta\left(G_1^2 + \lambda^2 G_2^2\right) + \zeta_t(\mathbf{x}),\end{aligned}$$

where $\zeta_t(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_t)^\top(\mathbf{g}(\mathbf{x}_t) - \nabla f(\mathbf{x}_t))$. Furthermore by the convexity of $F(\mathbf{x})$, we have

$$\begin{aligned}F(\mathbf{x}_t) - F(\mathbf{x}) \leq\ & \frac{1}{2\eta}\left(\|\mathbf{x} - \mathbf{x}_t\|_2^2 - \|\mathbf{x} - \mathbf{x}_{t+1}\|_2^2\right) + \\ & \eta(G_1^2 + \lambda^2 G_2^2) + \zeta_t(\mathbf{x}).\end{aligned}$$

Noting that $\mathrm{E}_t[\zeta_t(\mathbf{x})] = 0$, taking expectation over randomness and summation over $t = 1, \ldots, T$, we have

$$\begin{aligned}\frac{1}{T}\mathrm{E}\left[\sum_{t=1}^{T}(F(\mathbf{x}_t) - F(\mathbf{x}))\right] &= \mathrm{E}\left[F(\widehat{\mathbf{x}}_t) - F(\mathbf{x})\right] \\ &\leq \frac{\mathrm{E}[\|\mathbf{x}_1 - \mathbf{x}\|_2^2]}{2\eta T} + \eta(G_1^2 + \lambda^2 G_2^2).\end{aligned}$$

Let $B = \mathrm{E}[\|\mathbf{x}_1 - \mathbf{x}_*\|_2^2]/(2\eta T) + \eta(G_1^2 + \lambda^2 G_2^2)$. Since $\mathbf{x}_* \in \mathcal{D} \subseteq \mathcal{B}$, we have

$$\mathrm{E}\left[F(\widehat{\mathbf{x}}_t) - F(\mathbf{x}_*)\right] \leq B. \tag{8}$$

It follows that

$$\mathrm{E}[f(\widehat{\mathbf{x}}_T) + \lambda[c(\widehat{\mathbf{x}}_T)]_+] \leq f(\mathbf{x}_*) + B. \tag{9}$$

If $c(\widehat{\mathbf{x}}_T) \leq 0$, we have $\widetilde{\mathbf{x}}_T = \widehat{\mathbf{x}}_T$. Following from $F(\widetilde{\mathbf{x}}_T) \geq f(\widetilde{\mathbf{x}}_T)$ and $F(\mathbf{x}_*) = f(\mathbf{x}_*)$, we can verify that $\mathrm{E}[f(\widehat{\mathbf{x}}_T)] - f(\mathbf{x}_*) \leq B$ and also $\mathrm{E}[f(\widehat{\mathbf{x}}_T)] - f(\mathbf{x}_*) \leq \rho B/(\rho - G_1/\lambda)$ holds.

Next we show that $\mathrm{E}[f(\widehat{\mathbf{x}}_T)] - f(\mathbf{x}_*) \leq \rho B/(\rho - G_1/\lambda)$ holds when $c(\widehat{\mathbf{x}}_T) > 0$. From Lemma 1, we have

$$c(\widehat{\mathbf{x}}_T) \geq \rho\|\widehat{\mathbf{x}}_T - \widetilde{\mathbf{x}}_T\|_2. \tag{10}$$

Moreover it follows from $\|\partial f(\mathbf{x})\|_2 \leq G_1$ and $f(\mathbf{x}_*) \leq f(\widetilde{\mathbf{x}}_T)$ that the following inequality holds

$$\begin{aligned}f(\mathbf{x}_*) - f(\widehat{\mathbf{x}}_T) &\leq f(\mathbf{x}_*) - f(\widetilde{\mathbf{x}}_T) + f(\widetilde{\mathbf{x}}_T) - f(\widehat{\mathbf{x}}_T) \\ &\leq G_1\|\widehat{\mathbf{x}}_T - \widetilde{\mathbf{x}}_T\|_2.\end{aligned} \tag{11}$$

Substituting Eqs. (10) and (11) into Eq. (9), we have

$$\begin{aligned}\lambda\rho\mathrm{E}[\|\widehat{\mathbf{x}}_T - \widetilde{\mathbf{x}}_T\|_2] &\leq \mathrm{E}[f(\mathbf{x}_*) - f(\widehat{\mathbf{x}}_T)] + B \\ &\leq G_1\mathrm{E}[\|\widehat{\mathbf{x}}_T - \widetilde{\mathbf{x}}_T\|_2] + B.\end{aligned}$$

By some rearrangement, we have $\mathrm{E}[\|\widehat{\mathbf{x}}_T - \widetilde{\mathbf{x}}_T\|_2] \leq B/(\lambda\rho - G_1)$. Furthermore we have

$$\begin{aligned}&\mathrm{E}[f(\widetilde{\mathbf{x}}_T)] - f(\mathbf{x}_*) \leq \mathrm{E}[f(\widetilde{\mathbf{x}}_T) - f(\widehat{\mathbf{x}}_T)] + \mathrm{E}[f(\widehat{\mathbf{x}}_T)] \\ &-f(\mathbf{x}_*) \leq \mathrm{E}[G_1\|\widehat{\mathbf{x}}_T - \widetilde{\mathbf{x}}_T\|_2] + B \leq \frac{\lambda\rho}{\lambda\rho - G_1}B,\end{aligned}$$

where the second inequality follows from $\|\nabla f(\mathbf{x})\|_2 \leq \mathrm{E}\|\nabla f(\mathbf{x}; \varepsilon)\| \leq G_1$, and $|f(\mathbf{x}) - f(\mathbf{y})| \leq G_1\|\mathbf{x} - \mathbf{y}\|_2$ for any $\mathbf{x}, \mathbf{y}$. This completes the proof of the lemma. □

We present a main convergence result of the Epro-SGD algorithm in the following theorem.

Theorem 1. *Under Assumptions A1$\sim$A3 and given that $f(\mathbf{x})$ is β-strongly convex, if we let $\mu = \rho/(\rho - G_1/\lambda)$, $G^2 = G_1^2 + \lambda^2 G_2^2$, and set $T_1 = 8, \eta_1 = \mu/(2\beta)$, the total number of epochs $k^\dagger$ in Algorithm 1 is given by*

$$k^\dagger = \left\lceil \log_2\left(\frac{T}{8} + 1\right)\right\rceil \leq \log_2\left(\frac{T}{4}\right), \tag{12}$$

the solution $\mathbf{x}_1^{k^\dagger+1}$ enjoys a convergence rate of

$$\mathrm{E}[f(\mathbf{x}_1^{k^\dagger+1})] - f(\mathbf{x}_*) \leq \frac{32\mu^2 G^2}{\beta(T+8)}, \tag{13}$$

and $c(\mathbf{x}_1^{k^\dagger+1}) \leq 0$.

Proof. From the updating rule $T_{k+1} = 2T_k$, we can easily verify Eq. (12). Since $\mathbf{x}_1^{k^\dagger+1} = \widetilde{\mathbf{x}}_T^{k^\dagger} \in \mathcal{D}$, the inequality $c(\mathbf{x}_1^{k^\dagger+1}) \leq 0$ trivially holds.

Let $V_k = \mu^2 G^2 / \left(2^{k-2}\beta\right)$. It follows that $T_k = 2^{k+2} = 16\mu^2 G^2 / \left(V_k \beta\right)$ and $\eta_k = \mu / \left(2^k \beta\right) = V_k / \left(4\mu G^2\right)$. Next we show the inequality

$$\mathrm{E}[f(\mathbf{x}^k)] - f(\mathbf{x}_*) \leq V_k \tag{14}$$

holds by induction. Note that Eq. (14) implies $\mathrm{E}[f(\mathbf{x}_1^{k+1})] - f(\mathbf{x}_*) \leq 32\mu^2 G^2 / \left(\beta(T+8)\right)$, due to $V_k < 32\mu^2 G^2 / \left(\beta(T+8)\right)$. Let $\Delta_k = f(\mathbf{x}_1^k) - f(\mathbf{x}_*)$. It follows from Lemma 5, $\mu > 1$, and $G^2 > G_1^2$, the inequality in Eq. (14) holds when $k = 1$. Assuming that Eq. (14) holds for $k = k^\dagger$, we show that Eq. (14) holds for $k = {k^\dagger}_1 + 1$.

For a random variable X measurable with respect to the randomness up to epoch $k^\dagger + 1$. Let $\mathrm{E}_{k^\dagger}[X]$ denote the expectation conditioned on all the randomness up to epoch $k^\dagger$. Following Lemma 2, we have

$$\mathrm{E}_{k^\dagger}[\Delta_{k^\dagger+1}] \leq \mu \left[\eta_{k^\dagger} G^2 + \frac{\mathrm{E}[\|\mathbf{x}_1^{k^\dagger} - \mathbf{x}_*\|_2^2]}{2\eta_{k^\dagger} T_{k^\dagger}} \right].$$

Since $\Delta_{k^\dagger} = f(\mathbf{x}_1^{k^\dagger}) - f(\mathbf{x}_*) \geq \beta \|\mathbf{x}_1^{k^\dagger} - \mathbf{x}_*\|_2^2 / 2$ by the strong convexity in $f(\cdot)$, we have

$$\begin{aligned} \mathrm{E}[\Delta_{k^\dagger+1}] &\leq \mu \left[\eta_{k^\dagger} G^2 + \frac{\mathrm{E}[\Delta_{k^\dagger}]}{\eta_{k^\dagger} T_{k^\dagger} \beta} \right] \\ &= \mu \eta_{k^\dagger} G^2 + \frac{V_{k^\dagger} \mu}{\eta_{k^\dagger} T_{k^\dagger} \beta} = \frac{V_{k^\dagger}}{4} + \frac{V_{k^\dagger}}{4} = V_{k^\dagger+1}, \end{aligned}$$

which completes the proof of this theorem. □

Remark We compare the obtained main results in Theorem 1 with several existing works. Firstly Eq. (13) implies that Epro-SGD achieves an optimal bound $O(1/T)$, matching the lower bound for a strongly convex problem [15]. Secondly in contrast to the OneProj method [20] with a convergence rate $O(\log T / T)$, Epro-SGD uses no more than $\log_2(T/4)$ projections to obtain an $O(1/T)$ convergence rate. Epro-SGD thus has better control over the solution for not deviating (too much) from the feasibility domain in the intermediate iterations. Thirdly compared to Epoch-SGD with its convergence rate bounded by $O\left(8G_1^2 / \left(\beta T\right)\right)$, the convergence rate bound of Epro-SGD is only worse by a factor of constant $4\mu^2 G^2 / G_1^2$. Particularly consider a positive definite constraint with $\rho = 1$, $\mu = 2$, and $\lambda = 2G_1/\rho$, we have $G^2 = 5G_1^2$ and the bound of Epro-SGD is only worse by a factor of 80 than Epoch-SGD. Finally compared to the logT-SGD algorithm [33] which requires $O(\kappa \log_2 T)$ projections (κ is the conditional number), the number of projections in Epro-SGD is independent of the conditional number.

The main results in Lemma 2 and Theorem 1 are expected convergence bounds. In Theorem 2 (proof provided in Appendix) we show that Epro-SGD also enjoys a high probability bound under a boundedness assumption, i.e., $\|\mathbf{x}_* - \mathbf{x}_t\|_2 \leq D$ for all t. Note that the existing Epoch-SGD method [15] uses two different methods to derive its high probability bounds. Specifically the first method relies on an efficient function evaluator to select the best solutions among multiple trials of run; while the second one modifies the updating rule by projecting the solution onto the intersection of the domain and a center-shifted bounded ball with decaying radius. These two methods however may lead to additional computation steps, if being adopted for deriving high probability bounds for Epro-SGD.

Theorem 2. *Under Assumptions A1∼A3 and given $\|\mathbf{x}_t - \mathbf{x}_*\|_2 \leq D$ for all t. If we let $\mu = \rho/(\rho - G_1/\lambda)$, $G^2 = G_1^2 + \lambda^2 G_2^2, C = \left(8G_1^2/\beta + 2G_1 D\right)\ln(m/\epsilon) + 2G_1 D$, and set $T_1 \geq \max\left(3C\beta / \left(\mu G^2\right), 9\right)$, $\eta_1 = \mu/(3\beta)$, the total number of epochs $k^\dagger$ in Algorithm 1 is given by*

$$k^\dagger = \left\lfloor \log_2 \left(\frac{T}{T_1} + 1 \right) \right\rfloor \leq \log_2(T/4),$$

and the final solution $\mathbf{x}_1^{k^\dagger+1}$ enjoys a convergence rate of

$$f(\mathbf{x}_1^{k^\dagger+1}) - f(\mathbf{x}_*) \leq \frac{4T_1 \mu^2 G^2}{\beta(T + T_1)}$$

with a probability at least $1 - \delta$, where $m = \lceil 2 \log_2 T \rceil$.

Remark The assumption $\|\mathbf{x}_* - \mathbf{x}_t\|_2 \leq D$ can be satisfied practically, if we estimate the value of D such that $\|\mathbf{x}_*\|_2 \leq D/2$, and then project the intermediate solutions onto $\|\mathbf{x}\|_2 \leq D/2$ at every iteration. Note that Epoch-SGD [15] requires a total number of T projections, and its high probability bound of Epoch-SGD is denoted by $f(\mathbf{x}_1^{k^\dagger+1}) - f(\mathbf{x}_*) \leq 1200 G_1^2 \log(1/\widetilde{\delta}) / \left(\beta T\right)$ with a probability at lest $1 - \delta$, where $\widetilde{\delta} = \delta / \left(\lfloor \log_2(T/300 + 1) \rfloor\right)$.

3.3 A PROXIMAL VARIANT

We propose a proximal extension of Epro-SGD, by exploiting the structure of the objective function. Let the objective function in Eq. (1) be a sum of two components

$$\widehat{f}(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}),$$

where $g(\mathbf{x})$ is a relatively simple function, for example a squared ℓ_2-norm or ℓ_1-norm, such that the involved proximal mapping

$$\min_{\mathbf{x} \in \mathbb{R}^d} \quad g(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \widehat{\mathbf{x}}\|_2^2$$

is easy to compute. The optimization problem in Eq. (1) can be rewritten as

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^d} \quad & f(\mathbf{x}) + g(\mathbf{x}) \\ s.t. \quad & c(\mathbf{x}) \leq 0. \end{aligned} \tag{15}$$

Denote by $\mathbf{x}_*$ the optimal solution to Eq. (15). We similarly introduce an augmented objective function as

$$F(\mathbf{x}) = f(\mathbf{x}) + \lambda[c(\mathbf{x})]_+ + g(\mathbf{x}). \tag{16}$$

The update of the proximal SGD method for solving (15) [9, 10, 22] is given by

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{D}} \frac{1}{2}\|\mathbf{x} - (\mathbf{x}_t - \eta\mathbf{g}(\mathbf{x}_t))\|_2^2 + \eta g(\mathbf{x}). \tag{17}$$

If $g(\mathbf{x})$ is a sparse regularizer, the proximal SGD can guarantee the sparsity in the intermediate solutions and usually yields better convergence than the standard SGD. However, given a complex constraint, solving the proximal mapping may be computational expensive. Therefore, we consider a proximal variant of Epro-SGD which involves only the proximal mapping of $g(\mathbf{x})$ without the constraint $\mathbf{x} \in \mathcal{D}$. An instinctive solution is to use the following update in place of step 6 in Algorithm 1:

$$\begin{aligned}\mathbf{x}_{t+1}^k = \arg\min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{x} - \\ [\mathbf{x}_t^k - \eta_k(\mathbf{g}(\mathbf{x}_t^k) + \lambda\partial[c(\mathbf{x}_t^k)]_+)]\|_2^2 + \eta_k g(\mathbf{x}). \end{aligned}\tag{18}$$

Based on this update and using techniques in Lemma 2, we obtain a similar convergence result (proof provided in Appendix), as presented in the following lemma [8].

Lemma 3. *Under Assumptions A1$\sim$A3 and setting $\mu = \rho/(\rho - G_1/\lambda)$, by applying the update in Eq. (18) a number of T iterations, we have*

$$\begin{aligned}\mathrm{E}[\widehat{f}(\widetilde{\mathbf{x}}_T^k)] - \widehat{f}(\mathbf{x}_*) \quad \leq \quad & \mu\mathrm{E}\left[\eta G^2 + \frac{\|\mathbf{x}_1^k - \mathbf{x}_*\|_2^2}{2\eta T}\right. \\ & \left. + \frac{g(\mathbf{x}_1^k) - g(\mathbf{x}_{T+1}^k)}{T}\right], \end{aligned}\tag{19}$$

where $G^2 = (G_1^2 + \lambda^2 G_2^2)$, and $\widetilde{\mathbf{x}}_T^k$ denotes the projected solution of the averaged solution $\widehat{\mathbf{x}}_T^k = \sum_{t=1}^T \mathbf{x}_t^k/T$.

Different from the main result in Lemma 2, Eq. (19) has an additional term $(g(\mathbf{x}_1^k) - g(\mathbf{x}_{T+1}^k))/T_k$; it makes the convergence analysis in Epro-SGD difficult. To overcome this difficulty, we adopt the optimal regularized dual averaging (ORDA) algorithm [6] for solving Eq. (16). The details of ORDA are presented in Algorithm 2. The main convergence results of ORDA are summarized in the following lemma (proof provided in Appendix).

Lemma 4. *Under Assumptions A1$\sim$A3 and setting $\mu = \rho/(\rho - G_1/\lambda)$, by running ORDA a number of T iterations for solving the augmented objective (16), we have*

$$\mathrm{E}[F(\widehat{\mathbf{x}}_T) - F(\mathbf{x}_*)] \leq \frac{4\|\mathbf{x}_1 - \mathbf{x}_*\|_2^2}{\eta\sqrt{T}} + \frac{2\eta(3G_1 + 2\lambda G_2)^2}{\sqrt{T}},$$

and

$$\begin{aligned}\mathrm{E}[\widehat{f}(\widetilde{\mathbf{x}}_T)] - \widehat{f}(\mathbf{x}_*) \quad \leq \quad & \mu\mathrm{E}\left[\frac{4\|\mathbf{x}_1 - \mathbf{x}_*\|_2^2}{\eta\sqrt{T}} + \right. \\ & \left. \frac{2\eta(3G_1 + 2\lambda G_2)^2}{\sqrt{T}}\right], \end{aligned}$$

where $\widetilde{\mathbf{x}}_T$ denotes the projected solution of the final solution $\widehat{\mathbf{x}}_T$.

Algorithm 2 Optimal Regularized Dual Averaging (ORDA)

```
Input: a step size $\eta$, the number iterations $T$, and the initial solution $\mathbf{x}_1$,
Set $\theta_t = \frac{2}{t+1}, \nu_t = \frac{2}{t}, \gamma_t = \frac{t^{3/2}}{\eta}$ and $\mathbf{z}_1 = \mathbf{x}_1$
for $t = 1, \ldots, T+1$ do
    compute $\mathbf{u}_t = (1-\theta_t)\mathbf{x}_t + \theta_t\mathbf{z}_t$
    compute a stochastic subgradient $\mathbf{g}(\mathbf{x}_t)$ of $f(\mathbf{x})$ at $\mathbf{x}_t$ and a subgradient of $[c(\mathbf{x}_t)]_+$
    let $\bar{\mathbf{g}}_t = \theta_t\nu_t\left(\sum_{\tau=1}^t \frac{\mathbf{g}(\mathbf{x}_\tau)+\lambda\partial[c(\mathbf{x}_\tau)]_+}{\nu_\tau}\right)$
    compute $\mathbf{z}_{t+1} = \arg\min_{\mathbf{x}} \bar{\mathbf{g}}_t^\top\mathbf{x} + \frac{\theta_t\nu_t\gamma_{t+1}}{2}\|\mathbf{x} - \mathbf{x}_1\|_2^2 + g(\mathbf{x})$
    compute $\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}} \mathbf{x}^\top(\mathbf{g}(\mathbf{x}_t) + \lambda\partial[c(\mathbf{x}_t)]_+) + \frac{\gamma_t}{2}\|\mathbf{x} - \mathbf{u}_t\|_2^2 + g(\mathbf{x})$
end for
Output: $\widehat{\mathbf{x}}_T = \mathbf{x}_{T+2}$
```

Algorithm 3 Epoch-projection ORDA (Epro-ORDA)

```
Input: an initial step size $\eta_1$, total number of iterations $T$, and number of iterations in the first epoch $T_1$, a Lagrangian multiplier $\lambda > G_1/\rho$
Initialization: $\mathbf{x}_1^1 \in \mathcal{D}$ and $k = 1$.
while $\sum_{i=1}^k T_i \leq T$ do
    Run ORDA to obtain $\widehat{\mathbf{x}}_T^k = \mathrm{ORDA}(\mathbf{x}_1^k, \eta_k, T_k)$
    Compute $\widetilde{\mathbf{x}}_T^k = \mathcal{P}_{\mathcal{D}}[\widehat{\mathbf{x}}_T^k]$
    Update $\mathbf{x}_1^{k+1} = \widetilde{\mathbf{x}}_T^k$, $T_{k+1} = 2T_k, \eta_{k+1} = \eta_k/\sqrt{2}$
    Set $k = k+1$
end while
```

We present a proximal variant of Epro-SGD, namely Epro-ORDA, in Algorithm 3, and summarize its convergence results in Theorem 3. Note that Algorithm 2 and the convergence analysis in Lemma 4 are independent of the strong convexity in $\widehat{f}(\mathbf{x})$; the strong convexity is however used for analyzing the convergence of Epro-ORDA in Theorem 3 (proof provided in Appendix).

Theorem 3. *Under Assumptions A1$\sim$A3 and given that $\widehat{f}(\mathbf{x})$ is β-strongly convex, if we let $\mu = \rho/(\rho - G_1/\lambda)$ and $G = 3G_1 + 2\lambda G_2$, and set $T_1 = 16, \eta_1 = \mu/\beta$, then the total number of epochs $k^\dagger$ in Algorithm 3 is given by*

$$k^\dagger = \left\lfloor \log_2\left(\frac{T}{17} + 1\right)\right\rfloor \leq \log_2(T/8),$$

and the final solution $\mathbf{x}_1^{k^\dagger+1}$ enjoys a convergence rate of

$$\mathrm{E}[\widehat{f}(\mathbf{x}_1^{k^\dagger+1})] - \widehat{f}(\mathbf{x}_*) \leq \frac{68\mu^2 G^2}{\beta(T+17)},$$

and $c(\mathbf{x}_1^{k^\dagger+1}) \leq 0$.

4 AN EXAMPLE OF SOLVING LMNN VIA EPRO-SGD

In this section, we discuss an application of applying the proposed Epro-SGD to solve a high dimensional distance metric learning (DML) with a large margin formulation, i.e., the large margin nearest neighbor (LMNN) classification method [30]. LMNN classification is one of the state-of-the-art methods for k-nearest neighbor classification. It learns a positive semi-definite distance metric, based on which the examples from the k-nearest neighbors always belong to the same class, while the examples from different classes are separated by a large margin.

To describe the LMNN method, we first present some notations. Let $(x_i, y_i), i = 1, 2, \cdots, \widehat{N}$, be a set of data points, where $x_i \in \mathbb{R}^d$ and $y \in \mathcal{Y}$ denote the feature representation and the class label, respectively. Let A be a positive definite matrix that defines a distance metric as $\text{dist}(x_1, x_2) = \|x_1 - x_2\|_A^2 = (x_1 - x_2)^\top A(x_1 - x_2)$. To learn a distance metric that separates the examples from different classes by a large margin, one needs to extract a set of similar examples (from the same class) and dissimilar examples (from a different class), denoted by $(x_1^j, x_2^j, x_3^j), j = 1, \ldots N$, where x_1^j shares the same class label to x_2^j and a different class from x_3^j. To this end, for each example $x_1^j = x_i$ one can form x_2^j by extracting the k nearest neighbors (defined by an Euclidean distance metric) that share the same class label to x_i, and form x_3^j by extracting a set of examples that have a different class label. Then an appropriate distance metric could be obtained from the following constrained optimization problem

$$\begin{aligned} &\min_A \frac{c}{N}\sum_{j=1}^N \ell\left(A, x_1^j, x_2^j, x_3^j\right) + (1-c)tr(AL) + \frac{\mu_1}{2}\|A\|_F^2 \\ &s.t.\ A \succeq \epsilon I, \end{aligned} \tag{20}$$

where $\ell(A, x_1^j, x_2^j, x_3^j) = \max(0, \|x_1^j - x_2^j\|_A^2 - \|x_1^j - x_3^j\|_A^2 + 1)$ is a hinge loss and $c \in (0, 1)$ is a trade-off parameter. In Eq. (20), $A \succeq \epsilon I$ is used as the constraint to ensure that Assumption A3 holds. Minimizing the first term is equivalent to maximizing the margin between $\|x_1^j - x_3^j\|_A^2$ and $\|x_1^j - x_2^j\|_A^2$. The matrix L encodes certain prior knowledge about the distance metric; for example, the original LMNN work [30] defines L as $L = \sum_{l=1}^m \|x_1^l - x_2^l\|_A^2/m$, where (x_1^l, x_2^l) are all k-nearest neighbor pairs from the same class. Other works [19] have used a weighted summation of distances between all data pairs $L = \sum_{i\neq j}^n w_{ij}\|x_i - x_j\|_A^2/n(n-1)$ or intra-class covariance matrix [25]. The last term $\|A\|_F^2/2$ is used as a regularization term and also makes the objective function strongly convex.

For data sets of very high dimensionality, i.e., $d \gg n$, LMNN in Eq. (20) usually produces a sub-optimal solution [25], as this formulation does not capture the sparsity structure of the features. Therefore we add a sparse regularizer and express the formulation below

$$\begin{aligned} \min_A \quad & \frac{c}{N}\sum_{j=1}^N \ell\left(A, x_1^j, x_2^j, x_3^j\right) + (1-c)tr(AL) \\ & + \frac{\mu_1}{2}\|A\|_F^2 + \mu_2\|A\|_1^{\text{off}} \\ s.t. \quad & A \succeq \epsilon I, \end{aligned} \tag{21}$$

where $\|A\|_1^{\text{off}} = \sum_{i\neq j}|A_{ij}|$ is an elmenent-wise ℓ_1-norm excluding the diagonal entries. Note that this sparse regularizer $\|A\|_1^{\text{off}}$ have been previously used in [25] for a different purpose.

Many standard optimization solvers or algorithms may not be efficient for solving Eq. (21). Firstly, the optimization problem in Eq. (21) can be formulated as a semi-definite program (SDP); however, general SDP solvers usually scale poorly with the number of triplets and is not suitable for large scale data analysis. Secondly, the gradient decent method presented in [30] requires to project intermediate solutions onto a positive definite cone; this operation invokes expensive singular value decomposition (SVD) for a large matrix and this limitation restricts the real-world applications of the gradient descent method. Thirdly, [25] employs a block coordinate descent (BCD) method to solve an L1-penalized log-det optimization problem; the BCD method is not suitable for solving Eq. (21), as the loss function is not linear in the variable A.

We employ the proposed Epro-SGD algorithm to solve the LMNN formulation in Eq. (21). Let $f(A) = \frac{c}{N}\sum_{j=1}^N \ell(A, x_1^j, x_2^j, x_3^j) + (1-c)tr(AL)$ and $g(A) = \frac{\mu_1}{2}\|A\|_F^2 + \mu_2\|A\|_1^{\text{off}}$. The positive definite constraint can be rewritten into an inequality constraint as $c(A) = \epsilon - \lambda_{\min}(A) \leq 0$, where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of the matrix A. We also make the correspondences $\mathbb{R}^d \to \mathbb{R}^{d\times d}$, $\mathbf{x} \to A$, $\|\mathbf{x}\|_2 \to \|A\|_F$, and provide necessary details below in a question-answer form

- How to compute the stochastic gradient of $f(A)$? First sample one triplet (x_1^j, x_2^j, x_3^j) (or a small number of triplets) and then compute $\widetilde{\nabla} f(A; \varepsilon) = c[(x_1^j - x_2^j)(x_1^j - x_2^j)^\top - (x_1^j - x_3^j)(x_1^j - x_3^j)^\top] + (1-c)L$ if $\ell(\|x_1^j - x_2^j\|_A^2 - \|x_1^j - x_3^j\|_A^2 + 1) > 0$, $\widetilde{\nabla} f(A; \varepsilon) = (1-c)L$ otherwise.

- How to compute the gradient of $[c(A)]_+ = [\epsilon - \lambda_{\min}(A)]_+$? By the theory of matrix analysis, the subgradient of $[c(A)]_+$ can be computed by $\partial c(A) = -\mathbf{u}\mathbf{u}^\top$ if $c(A) > 0$, and zero otherwise, where $\mathbf{u}$ denotes the eigevector of A associated with its minimum eigenvalue.

- What is the solution to the following proximal gradient step?

$$\min_{A} \frac{1}{2}\|A - \bar{A}_{t+1}\|_F^2 + \eta \left(\frac{\mu_1}{2}\|A\|_F^2 + \mu_2\|A\|_1^{\text{off}}\right).$$

The solution can be obtained via a soft-thresholding algorithm [2].

- What are the appropriate values for β, ρ, r, λ, that are necessary for running the algorithm? The value of $\beta = \mu_1$. The value of ρ is $\min_{c(A)=0} \|\nabla c(A)\|_F = 1$. The value of r can be set to $\sqrt{2c/\mu_1}$. The value of $G_2 = 1$. The value of G_1 can be estimated as $8cR^2 + (1-c)\|L\|_F + \mu_1 r + \mu_2 d$ if we assume $\|x_i\|_2 \le R, i = 1, \ldots, n$. The value of $\lambda > G_1$ is usually tuned among a set of prespecified values.

Finally, we discuss the impact of employing Epro-SGD and Epro-ORDA method on accelerating the computation for solving LMNN. Note that at each iteration to compute the gradient of $c(A)$, we need to compute the minimum eigenvalue and its eigen-vector. For a dense matrix, it usually involves a time complexity of $O(d^2)$. However, by employing a proximal projection, we can guarantee that the intermediate solution A_t is a element-wise sparse solution, for which the computation of the last eigen-pair can be substantially reduced to be linear to the number of non-zeros elements in A_t.

To analyze the running time compared to the Epoch-SGD method, let us assume we are interested in an ϵ-accurate solution. In the following discussion, we take a particular choice of $\lambda = 2G_1$ and suppress the dependence on constants and only consider dependence on T, G_1 and d. The number of iterations required by Epro-SGD or Epro-ORDA is $\Omega\left(G_1^2/\epsilon\mu_1\right)$. Taking into account the running time per iteration, the total running time of Epoch-SGD is $\Omega\left(G_1^2 d^3/\left(\epsilon\mu_1\right)\right)$ and that of Epro-SGD/Epro-ORDA is $\Omega\left(G_1^2 d^2/\left(\epsilon\mu_1\right)\right)$. When d is very large, the speed-up can be orders of magnitude.

5 EXPERIMENTS

In this section, we empirically demonstrate the efficiency and effectiveness of the proposed Epro-SGD algorithm. We compare the following four algorithms:

- Stochastic sub-Gradient Descent method (SGD) [27]: we set the step size $\eta_t = 1/(\lambda t)$ and SGD achieves a rate of convergence $O(logT/T)$, requiring $O(T)$ projections for a constrained convex optimization problem.

- One-Projection SGD method (OneProj) [20]: we set the step size $\eta_t = 1/(\lambda t)$ and OneProj achieves a rate of convergence $O(logT/T)$, requiring only one projection for a constrained strongly convex optimization problem.

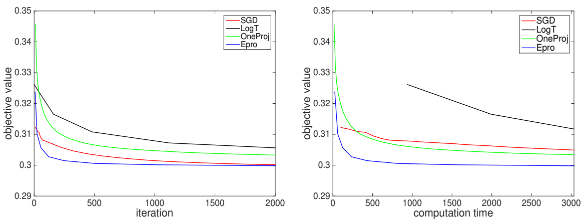

Figure 1: Empirical comparison of the four competing methods for solving Eq. (22). (1) Left plot: the change of the objective values with respect to the iteration number. (2) Right plot: the change of the objective values with respect to the computation time (in seconds).

- $O(logT)$-projections SGD method (logT) [33]: we set the step size $\eta_t = 1/(\sqrt{6}L)$ and logT achieves a rate of convergence $O(1/T)$, requiring $O(\log T)$ projections steps for a constrained strongly convex optimization problem.

- the proposed Epro-SGD with $O(logT)$ number of projections (Epro): we set the step size $\eta_t = 1/(\lambda t)$ and Epro archives a rate of convergence $O(1/T)$ with $O(\log T)$ projections steps for a constrained strongly convex optimization problem.

For illustration, we apply the competing algorithms for solving the constrained Lasso problem and the Large Margin Nearest Neighbor Classification (LMNN) in Eq. (21) respectively. We implement all algorithms using Matlab R2015a and conduct all simulations on an Intel(R) Xeon(R) CPU E5-2430 (15M Cache, 2.20 GHz).

5.1 EXPERIMENTS ON THE CONSTRAINED LASSO FORMULATION

We apply the proposed Epro-SGD algorithm and the other three competing algorithms to solve the L1-norm constrained least squares optimization problem

$$\begin{aligned} \min_{w} \quad & \frac{1}{2N}\sum_{i=1}^{N}\left(x_i^T w - y_i\right)^2 + \alpha\|w\|^2 \\ s.t. \quad & \|w\|_1 \le \beta. \end{aligned} \tag{22}$$

Eq. (22) is an equivalent constrained counterpart of the well studied Lasso formation [29]. They aim at achieving entry-wise sparsity in the weight vector w while computing a linear predictor for regression.

We use the algebra data, a benchmark data from KDD Cup 2010 [28], for the following experiments. Specifically we use a preprocessed version of the algebra data[1] for our simulations. This preprocessed data set consists of $8,407,752$

[1] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

samples from two classes, and each of the samples is represented as a feature feature of dimensionality $20,216,830$. In our experiments, we set $\alpha = 1$ and $\beta = 0.5$ for Eq. (22). We tune the initial step size respectively for each of the competing algorithms to get the (nearly) best performance; specifically in this experiments, we set $\eta_0 = 0.5$ for SGD, $\eta_0 = 0.3$ and $\lambda = 0.03$ for Epro, $\eta_0 = 0.1$ for OneProj, and $\eta_0 = 0.1$ for LogT.

In the experiments, we respectively run all competing algorithms for 2000 iterations; we then record the obtained objective values and the corresponding computation time. The experimental results are presented in Figure 1. The left plot shows how the objective value is changed with respect to the algorithm iteration. Note that for Eq. (22), the number of algorithm iterations is equal to the number of stochastic gradient computation (the access to the subgradient of the objective function). From this plot, we can observe that after running 2000 iterations, Epro and SGD attain smaller objective values, compared to OneProj and LogT; we can also observe that OneProj empirically converges slightly faster than logT. The right plot shows how the objective value is changed with respect to the computation time. For this experiment, we set the maximum computation time to 3035 seconds, which is the computation time required by running Epro for 2000 iterations. We can observe that Epro attain a smaller objective value, compared to the other three competing method; meanwhile, the standard SGD and OneProj attain similar objective values, given a fixed amount of computation time.

5.2 EXPERIMENTS ON THE LMNN FORMULATION

We apply the four competing algorithms to solve the LMNN formulation in Eq. (21). We use the Cora data [24] for the following experiments. Cora consists of 2708 scientific publications exclusively from 7 different categories. Each publication is represented by a normalized vector of length 1 and dimensionality 1433. From this data, we construct 5416 neighbor pairs (NP) by randomly selecting 2 publications of the same label; we then construct 16248 non-neighbor (NNT) by randomly selecting 3 non-neighbor publications (of a different label) for each of the NPs. Therefore, in each iteration of the SGD-type methods, we can use a NP and a NNT to construct a stochastic gradient for the optimization formulation. We set $c = 0.5$, $\mu_1 = 10^{-4}$, and $\mu_2 = 10^{-3}$ in Eq. (21). We terminate the algorithms when the iteration number is larger than $4,000$ or the relative change of the objective values in two iterations is smaller than 10^{-8}; we also record the obtained objective values, the required iteration number, and the computation time. Similarly we tune the initial step size respectively for the competing algorithms; specifically, we set $\eta_0 = 4 \times 10^{-8}$ for SGD, $\eta_0 = 10^{-5}$ and $\lambda = 0.1$ for Epro, $\eta_0 = 5 \times 10^{-7}$ for OneProj, and $\eta_0 = 10^{-6}$ for LogT.

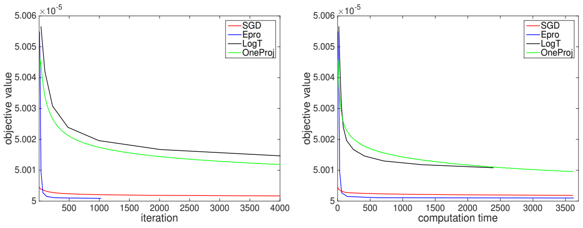

Figure 2: Empirical comparison of the four competing methods for solving Eq. (21). (1) Left plot: the change of the objective values with respect to the iteration number. (2) Right plot: the change of the objective value with respect to the computation time.

The experimental results are presented in Figure 2. Similarly in the left plot, we illustrate how the objective value changes with respect to the iteration number. For the LMNN formulation in Eq. (21), we can observe that Epro converges empirically much faster than all three competing algorithms; in particular, Epro converges after 1024 iterations, while the other 3 algorithms need more iterations. In the right plot, we illustrate how the objective value changes with respect to the computation time. We can observe that Epro converges using a smaller amount of computation time. specifically, in our experiment Epro converges with the computation time as 3622 seconds; while the other three competing algorithms need much more computation time.

6 CONCLUSIONS

We proposed an epoch-projection based SGD method, called Epro-SGD, for stochastic strongly convex optimization. The proposed Epro-SGD applies SGD on each iteration within its epochs and only performs a projection at the end of each epoch. Our analysis shows that Epro-SGD requires only a logarithmic number of projections, while achieves a guaranteed optimal rate of convergence both in expectation as well as with high probability. Additionally we proposed a variant of Epro-SGD based on an existing dual averaging method, called Epro-ORDA, which exploit structures of the optimization problems by incorporating an associated proximal mapping iteratively. For illustration, we applied the proposed Epro-SGD method for solving a large margin distance metric learning formulation and a constrained Lasso formulation respectively with a positive definite constraint. Our empirical results demonstrate the effectiveness of the proposed method.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments. T. Yang was supported in part by NSF (1463988, 1545995).

References

[1] Bach, F., Moulines, E.: Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In: NIPS (2011)

[2] Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences **2**(1), 183–202 (2009)

[3] Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: COMPSTAT (2010)

[4] Boucheron, S., Lugosi, G., Bousquet, O.: Concentration inequalities. Advanced Lectures on Machine Learning pp. 208–240 (2004)

[5] Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)

[6] Chen, X., Lin, Q., Pena, J.: Optimal regularized dual averaging methods for stochastic optimization. In: NIPS (2012)

[7] Clarkson, K.L.: Coresets, sparse greedy approximation, and the frank-wolfe algorithm. ACM Transactions on Algorithms **6**(4) (2010)

[8] Duchi, J., Singer, Y.: Efficient online and batch learning using forward backward splitting. Journal of Machine Learning Research **10**, 2899–2934 (2009)

[9] Duchi, J.C., Shalev-Shwartz, S., Singer, Y., Tewari, A.: Composite objective mirror descent. In: COLT (2010)

[10] Duchi, J.C., Singer, Y.: Efficient learning using forward-backward splitting. In: NIPS (2009)

[11] Frank, M., Wolfe, P.: An algorithm for quadratic programming. naval research logistics. Naval Research Logistics **3**(1-2), 95–110 (1956)

[12] Garber, D., Hazan, E.: A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. ArXiv:1301.4666 math.LG (2013)

[13] Gilpin, A., Peña, J., Sandholm, T.: First-order algorithm with log(1/epsilon) convergence for epsilon-equilibrium in two-person zero-sum games. Math. Program. **133**(1-2), 279–298 (2012)

[14] Hazan, E.: Sparse approximate solutions to semidefinite programs. In: LATIN, pp. 306–316 (2008)

[15] Hazan, E., Kale, S.: Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. Journal of Machine Learning Research - Proceedings Track **19**, 421–436 (2011)

[16] Hazan, E., Kale, S.: Projection-free online learning. In: ICML (2012)

[17] Jaggi, M.: Sparse convex optimization methods for machine learning. Ph.D. thesis, ETH Zurich (2011). DOI 10.3929/ethz-a-007050453

[18] Jaggi, M.: Revisiting frank-wolfe: Projection-free sparse convex optimization. In: ICML (2013)

[19] Liu, W., Tian, X., Tao, D., Liu, J.: Constrained metric learning via distance gap maximization. In: AAAI (2010)

[20] Mahdavi, M., Yang, T., Jin, R., Zhu, S., Yi, J.: Stochastic gradient descent with only one projection. In: NIPS (2012)

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

[22] Nesterov, Y.: Gradient methods for minimizing composite objective function. Mathematical Programming **140**(1), 125–161 (2013)

[23] Nesterov, Y.: Introductory lectures on convex optimization: A basic course. Springer Science & Business Media (2013)

[24] Prithviraj, S., Galileo, N., Mustafa, B., Lise, G.: Collective classification in network data. AI Magazine **29**(3) (2008)

[25] Qi, G.J., Tang, J., Zha, Z.J., Chua, T.S., Zhang, H.J.: An efficient sparse metric learning in high-dimensional space via l1-penalized log-determinant regularization. In: ICML (2009)

[26] Rakhlin, A., Shamir, O., Sridharan, K.: Making gradient descent optimal for strongly convex stochastic optimization. In: ICML (2012)

[27] Shalev-Shwartz, S., Singer, Y., Srebro, N.: Pegasos: Primal estimated sub-gradient solver for svm. In: ICML (2007)

[28] Stamper, J., Niculescu-Mizil, A., Ritter, S., Gordon, G., Koedinger, K.: Algebra I 2008-2009. Challenge data set from KDD Cup 2010 Educational Data Mining Challenge. Find it at http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp (2010)

[29] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B **58**(1), 267–288 (1996)

[30] Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research **10**, 207–244 (2009)

[31] Yang, T., Lin, Q.: Rsg: Beating sgd without smoothness and/or strong convexity. CoRR **abs/1512.03107** (2016)

[32] Ying, Y., Li, P.: Distance metric learning with eigenvalue optimization. Journal of Machine Learning Research **13**(1), 1–26 (2012)

[33] Zhang, L., Yang, T., Jin, R., He, X.: O(logT) projections for stochastic optimization of smooth and strongly convex functions. In: ICML (2013)

SUPPLEMENTAL MATERIALS

Lemma 5. *[15] If $f(\mathbf{x})$ is β-strongly convex and $\mathbf{x}_*$ denotes the optimal solution to $\min_{\mathbf{x}\in\mathcal{D}} f(\mathbf{x})$. For any $\mathbf{x} \in \mathcal{D}$, we have $f(\mathbf{x}) - f(\mathbf{x}_*) \leq 2G_1^2/\beta$.*

Proof. From Assumption A1, we have $\|\partial f(\mathbf{x})\|_2 \leq G_1$. Hence

$$f(\mathbf{x}) - f(\mathbf{x}_*) \leq G_1\|\mathbf{x} - \mathbf{x}_*\|_2.$$

Moreover from the strong convexity in $f(\cdot)$ we have

$$f(\mathbf{x}) - f(\mathbf{x}_*) \geq \frac{\beta}{2}\|\mathbf{x} - \mathbf{x}_*\|_2^2.$$

From the two inequalities above, we can easily verify that

$$\|\mathbf{x} - \mathbf{x}_*\|_2 \leq \frac{2G_1}{\beta}, \;\; f(\mathbf{x}) - f(\mathbf{x}_*) \leq \frac{2G_1^2}{\beta}.$$

This completes the proof. □

Proof of Theorem 2

The proof of Theorem 2 is based on an important result, as summarized in Lemma 6.

Lemma 6. *[20] Assume $\|\mathbf{x}_* - \mathbf{x}_t\|_2 \leq D$ for all t. Define $D_T = \sum_{t=1}^T \|\mathbf{x}_t - \mathbf{x}\|_2^2$ and $\Lambda_T = \sum_{t=1}^T \zeta_t(\mathbf{x})$. We have*

$$\begin{aligned}\Pr\left(\Lambda_T \leq 4G_1\sqrt{D_T \ln\frac{m}{\epsilon}} + 2G_1 D\ln\frac{m}{\epsilon}\right)& \\ +\Pr\left(D_T \leq \frac{D^2}{T}\right) &\geq 1-\epsilon,\end{aligned}$$

where $m = \lceil 2\log_2 T\rceil$ and $\sum_{t=1}^T \zeta_t(\mathbf{x}) = \sum_{t=1}^T(\nabla f(\mathbf{x}_t) - \mathbf{g}(\mathbf{x}_t))^\top(\mathbf{x} - \mathbf{x}_t)$.

Proof of Theorem 2 The proof below follows from techniques used in Lemma 2 and Theorem 1. Since $F(\mathbf{x})$ is β-strongly convex, we have

$$F(\mathbf{x}_t) - F(\mathbf{x}) \leq (\mathbf{x}_t - \mathbf{x})^\top \nabla F(\mathbf{x}_t) - \frac{\beta}{2}\|\mathbf{x} - \mathbf{x}_t\|_2^2.$$

Combining the above inequality with the inequality in (8) and taking summation over all $t = 1, \ldots, T$, we have

$$\begin{aligned}\sum_{t=1}^T(F(\mathbf{x}_t) - F(\mathbf{x})) &\leq \underbrace{\frac{\|\mathbf{x}_1 - \mathbf{x}\|_2^2}{2\eta} + \eta T(G_1^2 + \lambda^2 G_2^2)}_{BT} \\ &+ \sum_{t=1}^T \zeta_t(\mathbf{x}) - \frac{\beta}{2}D_T. \qquad (23)\end{aligned}$$

We substitute the bound in Lemma 6 into the above inequality with $\mathbf{x} = \mathbf{x}^*$. We consider two cases. In the first case, we assume $D_T \leq D^2/T$. As a result, we have

$$\begin{aligned}\sum_{t=1}^T \zeta_t(\mathbf{x}^*) &= \sum_{t=1}^T(\nabla f(\mathbf{x}_t) - \mathbf{g}(\mathbf{x}_t))^\top(\mathbf{x}^* - \mathbf{x}_t) \\ &\leq 2G_1\sqrt{TD_T} \leq 2G_1 D,\end{aligned}$$

which together with the inequality in (23) leads to the bound

$$\sum_{t=1}^T(F(\mathbf{x}_t) - F(\mathbf{x}^*)) \leq 2G_1 D + BT.$$

In the second case, we assume

$$\begin{aligned}\sum_{t=1}^T \zeta_t(\mathbf{x}^*) &\leq 4G_1\sqrt{D_T\ln\frac{m}{\epsilon}} + 4G_1\ln\frac{m}{\epsilon} \\ &\leq \frac{\beta}{2}D_T + \left(\frac{8G_1^2}{\beta} + 4G_1\right)\ln\frac{m}{\epsilon},\end{aligned}$$

where the last step uses the fact $2\sqrt{ab} \leq a^2 + b^2$. We thus have

$$\sum_{t=1}^T(F(\mathbf{x}_t) - F(\mathbf{x}^*)) \leq \left(\frac{8G_1^2}{\beta} + 2G_1 D\right)\ln\frac{m}{\epsilon} + BT$$

Combing the results of the two cases, we have, with a probability $1-\epsilon$,

$$\begin{aligned}\sum_{t=1}^T(F(\mathbf{x}_t) - F(\mathbf{x}^*)) &\leq \left(\frac{8G_1^2}{\beta} + 2G_1 D\right)\ln\frac{m}{\epsilon} \\ &\quad + 2G_1 D + BT,\end{aligned}$$

where $C = \left(\frac{8G_1^2}{\beta} + 2G_1 D\right)\ln\frac{m}{\epsilon} + 2G_1 D$. Following the same analysis, we have

$$f(\widetilde{\mathbf{x}}_T) - f(\mathbf{x}_*) \leq \frac{\mu C}{T} + \frac{\mu\|\mathbf{x}_1 - \mathbf{x}_*\|_2^2}{2\eta T} + \mu\eta G^2$$

Let $\Delta_k = f(\mathbf{x}_k^1) - f(\mathbf{x}_*)$. By induction, we have

$$\Delta_{k+1} \leq \frac{\mu C}{T_k} + \frac{\mu\Delta_k}{2\eta_k T_k\beta} + \mu\eta_k G^2$$

Assume $\Delta_k \leq V_k \triangleq \frac{\mu^2 G^2}{2^{k-2}\beta}$, by plugging the values of η_k, T_k, we have

$$\Delta_{k+1} \leq \frac{V_k}{6} + \frac{V_k}{6} + \frac{V_k}{6} = \frac{V_k}{2} = V_{k+1}$$

where we use $T_1 \geq \max\left(\frac{3C\beta}{\mu G^2}, 9\right)$ and $T_k \geq \max\left(\frac{6\mu c}{V_k}, \frac{18\mu^2 G^2}{V_k\beta}\right)$ and $\eta_k = \frac{V_k}{6\mu G^2} = \frac{2\mu}{2^k(3\beta)}$. This completes the proof of this theorem.

Proof of Lemma 3

To prove Lemma 3, we derive an inequality similar to Eq. (8); the rest proof of Lemma 3 is similar to that of Lemma 2.

Corollary 1. *Given a β-strongly convex function $\widehat{f}(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$, and a sequence $\{\mathbf{x}_t\}$ defined by the update $\mathbf{x}_{t+1} = \min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - (\mathbf{x}_t - \eta\mathbf{g}(\mathbf{x}_t))\|_2^2 + \eta g(\mathbf{x})$. Then for any $\mathbf{x}$, we have*

$$\begin{aligned}
&\sum_{t=1}^{T} [f(\mathbf{x}_t) + g(\mathbf{x}_{t+1}) - f(\mathbf{x}) - g(\mathbf{x})] \\
\leq\ & \frac{\|\mathbf{x} - \mathbf{x}_1\|_2^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \|\mathbf{g}(\mathbf{x}_t)\|_2^2 + \sum_{t=1}^{T}(\mathbf{x} - \mathbf{x}_t)^\top(\mathbf{g}(\mathbf{x}_t) \\
& -\nabla f(\mathbf{x}_t)) - \frac{\beta}{2}\sum_{t=1}^{T} \|\mathbf{x} - \mathbf{x}_{t+1}\|_2^2.
\end{aligned}$$

Corollary 1 can be proved using techniques similar to the ones in [9] but with extra care on the stochastic gradient. As a consequence we have

$$\begin{aligned}
&\frac{1}{T}\mathrm{E}\left[\sum_{t=1}^{T} \hat{f}(\mathbf{x}_t) - \hat{f}(\mathbf{x})\right] \\
\leq\ & \frac{\mathrm{E}[\|\mathbf{x} - \mathbf{x}_1\|_2^2]}{2\eta T} + \eta(G_1^2 + \lambda G_2^2) + \frac{g(\mathbf{x}_1) - g(\mathbf{x}_{T+1})}{T}
\end{aligned}$$

Proof of Lemma 4

The lemma is a corollary of results in [6] for general convex optimization. In particular, if we consider the stochastic composite optimization

$$F(\mathbf{x}) = \phi(\mathbf{x}) + g(\mathbf{x})$$

where $g(\mathbf{x})$ is a simple function such that its proximal mapping can be easily solved and $\phi(\mathbf{x})$ is only accessible through a stochastic oracle that returns a stochastic subgradient $\mathbf{g}(\mathbf{x})$. To state the convergence of ORDA for general convex problems, [6] makes the following assumptions: (i) $\mathrm{E}[\|\mathbf{g}(\mathbf{x}) - \mathrm{E}\mathbf{g}(\mathbf{x})\|_2^2] \leq \sigma^2$ and (ii)

$$\phi(\mathbf{y}) - \phi(\mathbf{x}) - (\mathbf{y} - \mathbf{x})^\top \partial\phi(\mathbf{x}) \leq M\|\mathbf{y} - \mathbf{x}\|_2$$

When $\|\partial\phi(\mathbf{x})\|_2 \leq G$, the first inequality holds $\sigma = G$ and the second inequality holds with $M = 2G$. Applying to the augmented objective

$$F(\mathbf{x}) = f(\mathbf{x}) + \lambda[c(\mathbf{x})]_+ + g(\mathbf{x})$$

We note that $\sigma = G_1$ and $M = 2(G_1 + \lambda G_2)$. Follow the inequality (26) in the appendix of [6], we obtain that

$$\mathrm{E}[F(\mathbf{x}_{T+2}) - F(\mathbf{x}_*)] \leq \frac{4\|\mathbf{x}_1 - \mathbf{x}_*\|_2^2}{\eta\sqrt{T}} + \frac{2\eta(\sigma + M)^2}{\sqrt{T}}$$

by using the Euclidean distance $V(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$ and their notation $\tau = 1$, and noting that η is the inverse of their notation c. Then the second inequality is Lemma 4 can be proved similarly as for Lemma 2.

Proof of Theorem 3

Proof. Recall $\mu = \rho/(\rho - G_1/\lambda)$ and $G = 3G_1 + 2\lambda G_2$. Let $V_k = \left(\mu^2 G^2\right) / \left(2^{k-2}\beta\right)$. By the values of η_k and T_k we have

$$T_k = 2^{k+3} = \frac{32\mu^2 G^2}{V_k \beta}, \eta_k = \frac{\mu}{2^{(k-1)/2}\beta} = \frac{V_k\sqrt{T_k}}{8\mu G^2}.$$

Define $\Delta_k = \hat{f}(\mathbf{x}_1^k) - \hat{f}(\mathbf{x}_*)$. We first prove the inequality

$$\mathrm{E}[\Delta_k] \leq V_k$$

by induction. It is true for $k = 1$ because of Lemma 5, $\mu > 1$ and $G^2 > G_1^2$. Now assume it is true for k and we prove it for $k{+}1$. For a random variable X measurable with respect to the randomness up to epoch $k + 1$. Let $\mathrm{E}_k[X]$ denote the expectation conditioned on all the randomness up to epoch k. Following Lemma 2, we have

$$\mathrm{E}_k[\Delta_{k+1}] \leq \mu\left[\frac{2\eta_k G^2}{\sqrt{T_k}} + \frac{\mathrm{E}[4\|\mathbf{x}_1^k - \mathbf{x}_*\|_2^2]}{\eta_k\sqrt{T_k}}\right] \qquad (24)$$

Since $\Delta_k = f(\mathbf{x}_1^k) - f(\mathbf{x}_*) \geq \beta\|\mathbf{x}_1^k - \mathbf{x}_*\|_2^2/2$ by the strong convexity, we have

$$\begin{aligned}
\mathrm{E}[\Delta_{k+1}] &\leq \mu\left[\frac{2\eta_k G^2}{\sqrt{T_k}} + \frac{\mathrm{E}[8\Delta_k]}{\eta_k\sqrt{T_k}\beta}\right] \\
&= \frac{2\eta_k\mu G^2}{\sqrt{T_k}} + \frac{V_k\mu}{\eta_k\sqrt{T_k}\beta} = \frac{V_k}{4} + \frac{V_k}{4} = \frac{V_k}{2}
\end{aligned}$$

where we use the fact $\eta_k/\sqrt{T_k} = V_k/(8\mu G^2)$ and $T_k = 32\mu^2 G^2/(V_k\beta)$. Thus, we get

$$\mathrm{E}[f(\mathbf{x}_1^{k^\dagger+1})] - f(\mathbf{x}_*) = \mathrm{E}[\Delta_{k^\dagger+1}] \leq V_{k^\dagger+1} = \frac{\mu^2 G^2}{2^{k^\dagger-1}\beta}$$

Note that the total number of epochs satisfies

$$\sum_{k=1}^{k^\dagger}(T_k + 1) = 16(2^{k^\dagger} - 1) + k^\dagger \leq T$$

By some reformulations, we complete the proof of this theorem. □

Proof of Lemma 6

The proof of Lemma 6 is based on *the Bernstein Inequality for Martingales* [4]. We present its main result below for completeness.

Theorem 4. *[Bernstein Inequality for Martingales] Let $X_1, \ldots, X_n$ be a bounded martingale difference sequence with respect to the filtration $\mathcal{F} = (\mathcal{F}_i)_{1 \leq i \leq n}$ and with $\|X_i\| \leq K$. Let*

$$S_i = \sum_{j=1}^{i} X_j$$

be the associated martingale. Denote the sum of the conditional variances by

$$\Sigma_n^2 = \sum_{t=1}^{n} \mathrm{E}\left[X_t^2 | \mathcal{F}_{t-1}\right],$$

Then for all constants t, $\nu > 0$,

$$\Pr\left[\max_{i=1,\ldots,n} S_i > t \text{ and } \Sigma_n^2 \leq \nu\right] \leq \exp\left(-\frac{t^2}{2(\nu + Kt/3)}\right),$$

and therefore,

$$\Pr\left[\max_{i=1,\ldots,n} S_i > \sqrt{2\nu t} + \frac{\sqrt{2}}{3} Kt \text{ and } \Sigma_n^2 \leq \nu\right] \leq e^{-t}.$$

Proof of Lemma 6. Define martingale difference $X_t = (\mathbf{x} - \mathbf{x}_t)^\top (\nabla f(\mathbf{x}_t) - \mathbf{g}(\mathbf{x}_t))$ and martingale $\Lambda_T = \sum_{t=1}^{T} X_t$. Define the conditional variance Σ_T^2 as

$$\Sigma_T^2 = \sum_{t=1}^{T} \mathrm{E}_{\xi_t}\left[X_t^2\right] \leq 4G_1^2 \sum_{t=1}^{T} \|\mathbf{x}_t - \mathbf{x}\|_2^2 = 4G_1^2 D_T.$$

Define $K = 2G_1 D$. Thus, $\|X_t\|_2 \leq K$. We have

$$\begin{aligned}
& \Pr\left(\Lambda_T \geq 2\sqrt{4G_1^2 D_T \tau} + \sqrt{2}K\tau/3\right) \\
= & \Pr\left(\Lambda_T \geq 2\sqrt{4G_1^2 D_T \tau} + \sqrt{2}K\tau/3, \Sigma_T^2 \leq 4G_1^2 D_T\right) \\
= & \Pr\left(\Lambda_T \geq 2\sqrt{4G_1^2 D_T \tau} + \sqrt{2}K\tau/3, \Sigma_T^2 \leq 4G_1^2 D_T,\right. \\
& \left. D_T \leq \frac{D^2}{T}\right) + \sum_{i=1}^{m} \Pr\left(\Lambda_T \geq 2\sqrt{4G_1^2 D_T \tau} + \right. \\
& \left. \sqrt{2}K\tau/3, \Sigma_T^2 \leq 4G_1^2 D_T, \frac{D^2}{T} 2^{i-1} < D_T \leq \frac{D^2}{T} 2^i\right) \\
\leq & \Pr\left(D_T \leq \frac{D^2}{T}\right) + \sum_{i=1}^{m} \Pr\left(\Lambda_T \geq \sqrt{2 \times 4G_1^2 \frac{D^2}{T} 2^i \tau}\right. \\
& \left. + \sqrt{2}K\tau/3, \Sigma_T^2 \leq 4G_1^2 \frac{D^2}{T} 2^i\right) \\
\leq & \Pr\left(D_T \leq \frac{D^2}{T}\right) + me^{-\tau},
\end{aligned}$$

where we use the fact $\|\mathbf{x}_t - \mathbf{x}\|_2^2 \leq D^2$ for all t and $m = \lceil 2 \log_2 T \rceil$, and the last step follows the Bernstein inequality for martingales. We complete the proof by setting $\tau = \ln(m/\epsilon)$.