Level-Set Methods for Finite-Sum Constrained Convex Optimization

Qihang Lin [1] Runchao Ma [1] Tianbao Yang [2]

Abstract

We consider the constrained optimization where the objective function and the constraints are defined as summation of finitely many loss functions. This model has applications in machine learning such as Neyman-Pearson classification. We consider two level-set methods to solve this class of problems, an existing inexact Newton method and a feasible level-set method. To update the level parameter towards the optimality, both methods require an oracle that generates upper and lower bounds as well as an affine-minorant of the level function. To construct the desired oracle, we reformulate the level function as the value of a saddle-point problem using the conjugate and perspective of the loss functions. Then a stochastic variance-reduced gradient method with a special Bregman divergence is proposed as the oracle for solving that saddle-point problem. The special divergence ensures the proximal mapping in each iteration can be solved in a closed form. The total complexity of both level-set methods using the proposed oracle are analyzed.

1. Introduction

Constrained optimization arises in many fields of science and engineering and has been studied in a large volume of literature from both algorithmic or theoretical aspects (Bertsekas, 2014; 1999; Nocedal & Wright, 2006; Ruszczyński, 2006, and references therein). A general convex optimization problem with inequality constraints is formulated as

$$f^* := \min_{\mathbf{x}\in\mathcal{X}} f_0(\mathbf{x}) \quad \text{s.t.} \quad f_i(\mathbf{x}) \le r_i,\ i=1,2,\dots,m, \quad (1)$$

where $\mathcal{X} \subset \mathbb{R}^d$ is a closed convex set and f_i for $i = 0,\dots,m$ are closed convex real functions defined on $\mathcal{X}$. A solution $\bar{\mathbf{x}} \in \mathcal{X}$ is *ε-optimal* if $f_0(\bar{\mathbf{x}}) - f^* \le \varepsilon$ and *ε-feasible* if $\max_{i=1,\dots,m}[f_i(\bar{\mathbf{x}}) - r_i] \le \varepsilon$.

In this paper, we consider an important case of (1) where each f_i for $i = 0,\dots,m$ in (1) are given as a summation of a finite but large number of functions, i.e.,

$$f_i(\mathbf{x}) \quad = \quad \frac{1}{n_i}\sum_{j=1}^{n_i} f_{ij}(\mathbf{x}), \quad (2)$$

for $i = 0,\dots,m$, where f_{ij} are closed convex functions of $\mathbf{x} \in \mathcal{X}$ and n_i denotes the number of summands in f_i. An important case of (2) is the empirical risk minimization problem with empirical risk constraints where

$$f_{ij}(\mathbf{x}) = \phi_{ij}(\mathbf{x}^\top\xi_{ij}), j = 1,\dots,n_i, i = 0,\dots,m. \quad (3)$$

Here, $\phi_{ij}(z) : \mathbb{R} \to \mathbb{R}$ is a convex loss functions measuring the loss of the linear prediction $\mathbf{x}^\top\xi_{ij}$ on a data point ξ_{ij}.

Problem (1) with the structures given in (2) and (3) has many applications including multi-objective optimization (Mahdavi et al., 2013; Barba-Gonzaléz et al., 2017; Marler & Arora, 2004), shape-restricted regression (Seijo et al., 2011; Sen & Meyer, 2017; Lim, 2014; Fard et al., 2016; Cotter et al., 2016), and classification in Neyman-Pearson paradigm (Tong et al., 2016; Rigollet & Tong, 2011; Tong, 2013; Zhao et al., 2015).

One concrete example of (1) with $m = 1$ is Neyman-Pearson binary classification. Suppose the training data for a binary classification has been partitioned into the positive set $\{\xi_{0j}\}_{j=1}^{n_0}$ and the negative set $\{\xi_{1j}\}_{j=1}^{n_1}$. Neyman-Pearson classification problem is formulated as

$$\min_{\|\mathbf{x}\|_2\le\lambda} \frac{1}{n_0}\sum_{j=1}^{n_0}\phi(\mathbf{x}^\top\xi_{0j}),\ \text{s.t.}\ \frac{1}{n_1}\sum_{j=1}^{n_1}\phi(-\mathbf{x}^\top\xi_{1j}) \le r_1. \quad (4)$$

Here, $\|\mathbf{x}\|_2 \le \lambda$ is the constraint to avoid overfitting, ϕ can be a convex loss function, e.g, $\phi(z) = \log(1 + \exp(-z))$, and $r_1 > 0$ is a risk level. Minimizing the objective function can reduce the rate of Type-II error (identifying a positive instance as negative) while the constraint controls the rate of Type-I error (identifying a negative instance as positive) in a low level (less than r_1).

The following assumptions are made in the whole paper.

[1] Management Sciences Department, University of Iowa, Iowa City, IA, USA [2] Computer Science Department, University of Iowa, Iowa City, IA, USA. Correspondence to: Qihang Lin <qihang-lin@uiowa.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

Assumption 1. $\max_{i=1,\ldots,m}[f_i(\bar{\mathbf{x}}) - r_i] < 0$ *for some* $\bar{\mathbf{x}} \in \mathcal{X}$.

Assumption 2. *The set $\mathcal{X}$ is compact and either (a) ϕ_{ij} is smooth and its gradient is $\frac{1}{\gamma}$-Lipschitz continuous or (b) the domain of ϕ_{ij}^*, the Fenchel conjugate of ϕ_{ij}, is compact.*

Assumption 1 requires strictly feasiblity. Assumption 2 is satisfied by many commonly used loss functions. For example, the smooth hinge loss and logistic satisfy Assumption 2 (a) while the hinge loss satisfies Assumption 2 (b). To efficiently solve (1) with the structures (2) and (3) for large d and n_i, we consider first-order optimization algorithms, which are actively studied in the past decade due to their good scalability and easy implementation.

Contributions: Our main contributions in this paper are summarized as follows.

- We proposed an affine-minorized feasible level-set (AM-FLS) method, which is a new variant of the feasible level-set method in Lin et al. (2017). Compared to Lin et al. (2017) that guarantees a *relative* ε-optimal solution, our method utilizes an affine-minorant oracle (Definition 1) for a tighter estimation of the level function and guarantees an *absolute* ε-optimal solution.

- We compare the AM-FLS method with an inexact Newton level-set (IN-LS) method (Aravkin et al., 2016) under the same oracle. We show that both method have similar total complexity but the IN-LS method guarantee ε-feasibility when it terminates while the AM-FLS method produces a feasible solution path.

- To construct an efficient affine-minorant oracle for both level-set methods, we first provide a novel saddle-point reformulation of the level function using convex conjugate and perspective of the loss functions. Then, we design the oracle as a stochastic variance-reduced gradient (SVRG) method to solve this saddle-point problem using a special Bregman divergence. The special Bregmen divergence allows the proximal mapping in the SVRG method solved efficiently in a closed form. The complexity of both level-set methods with our affine-minorant oracle is $\tilde{O}(nd + \frac{n+d}{\varepsilon^2})$ where $n = \sum_{i=0}^m n_i$ while the existing deterministic oracle costs $\tilde{O}(\frac{nd}{\varepsilon})$.[1]

2. Two Level-Set Methods

The *level-set function* (Lemaréchal et al., 1995; Nesterov, 2013) for (1) is defined as

$$H(r) := \min_{\mathbf{x}\in\mathcal{X}} \mathcal{P}(r;\mathbf{x}) \tag{5}$$

where r is a *level parameter* and

$$\mathcal{P}(r;\mathbf{x}) := \max\left\{f_0(\mathbf{x}) - r, f_1(\mathbf{x}) - r_1, \ldots, f_m(\mathbf{x}) - r_m\right\}.$$

The following lemma contains the main properties of $H(r)$, which are collected from Lemmas 2.3.4, 2.3.5, and 2.3.6 in (Nesterov, 2004) and Lemma 1 in Lin et al. (2017).

Lemma 1. *It holds that*

(a) $H(r)$ *is non-increasing and convex in* r*;*

(b) $H(f^*) = 0$*;*

(c) $H(r) > 0$*, if* $r < f^*$ *and* $H(r) < 0$*, if* $r > f^*$*;*

(d) $H(r) - \delta \leq H(r+\delta) \leq H(r)$ *for any* $\delta \geq 0$.

According to this lemma, f^* is the unique root of $H(r)$. A *level-set method* solving (1) will generate a sequence of level parameters $r^{(1)}, r^{(2)}, \ldots$ approaching f^*, for example, by a root finding procedure. When a level parameter $\bar{r} \approx f^*$ is found, an approximate solution $\bar{\mathbf{x}} \approx \arg\min_{\mathbf{x}\in\mathcal{X}} \mathcal{P}(\bar{r};\mathbf{x})$ can be computed as an approximate solution of (1). The quality of $\bar{\mathbf{x}}$ can be justified in two situations. First, when $\bar{r} \leq f^*$ and $\mathcal{P}(\bar{r};\bar{\mathbf{x}}) \leq \varepsilon$, it is easy to see that $\bar{\mathbf{x}}$ is an ε-optimal and ε-feasible solution for (1). Second, when $f^* < \bar{r} \leq f^* + \varepsilon$ and $\mathcal{P}(\bar{r};\bar{\mathbf{x}}) \leq 0$, one can show that $\bar{\mathbf{x}}$ is an ε-optimal and feasible solution for (1).

Applying a classical root-finding algorithm to $H(r)$ to find $\bar{r} \approx f^*$ requires knowing the exact value of $H(r)$, which is hard to compute due to the nontrivial minimization in (5). One natural approach is to use an iterative optimization algorithm as an oracle to solve (5) approximately in order to get an upper bound $U(r)$ and a lower bound $L(r)$ of $H(r)$ and use them to update r towards f^*. For instance, by Lemma 1, we will know $r < f^*$ when we receive $L(r) > 0$ and $r > f^*$ when we receive $U(r) < 0$ from the oracle, which suggests if we should increase or decrease r. A class oracles used in a level-set method is defined below.

Definition 1. *An algorithm $\mathcal{A}(r, \theta, \varepsilon)$ is an* ***affine-minorant oracle*** *if, for any $\theta > 1$ and r, it returns $(L(r), U(r), S(r)) \in \mathbb{R}^3$ and $\bar{\mathbf{x}} \in \mathcal{X}$ such that:*

1. $L(r) \leq H(r) \leq U(r)$*;*
2. $\mathcal{P}(r;\bar{\mathbf{x}}) \leq U(r)$*;*
3. *Either* $\varepsilon < U(r) \leq \theta L(r)$ *or* $U(r) \leq \varepsilon$*, if* $r \leq f^*$*;*
4. $\theta U(r) \leq L(r)$*, if* $r > f^*$*;*
5. $H(r') \geq L(r) + S(r)(r' - r)$ *for any* r'.

Moreover, there exits a non-increasing function $\mathcal{C}(\cdot)$ such that the expected computational complexity[2] *for $\mathcal{A}$ to return these outputs is no more than $\mathcal{C}(\max\{|H(r)|, \varepsilon\})$ if $r \leq f^*$ and no more than $\mathcal{C}(|H(r)|)$ if $r > f^*$.*

[1] Here and in the rest of the paper, the logarithmic factors in complexity are suppressed in $\tilde{O}$.

[2] We consider expected complexity here because we allow $\mathcal{A}$ to be a stochastic algorithm that returns the desired output almost surely in a random complexity.

The name of affine-minorant oracle mainly comes from Property 5 above, which requires an affine function with a slope $S(r)$ that passes through $(r, L(r))$ and minorizes $H(r)$ globally. The original definition of affine-minorant oracle is given by Aravkin et al. (2016) for $r \leq f^*$. Here, we generalize it for any r in order to support the AM-FLS method where $r > f^*$. The inequality $H(r) \leq U(r)$ in Property 1 and Property 2 above can be easily satisfied with any $\bar{\mathbf{x}} \in \mathcal{X}$ by setting $U(r) = \mathcal{P}(r; \bar{\mathbf{x}})$, but not every optimization method can provide $L(r)$ and $S(r)$. Property 3 and 4 can be satisfied when $r \neq f^*$ as long as Property 1 holds and $U(r) - L(r) \leq \frac{\theta-1}{\theta}|H(r)|$. Typically, a smaller gap $U(r) - L(r)$ requires a higher computational cost in the oracle. Hence, the complexity of the oracle will increase (in a rate of $\mathcal{C}(|H(r)|)$) as $|H(r)|$ decreases to zero.

Before we specify the oracle, we first study the relationship between the complexity of an oracle and the complexity of solving (1) by level-set methods based on that oracle. We will focus on two different level-set methods: the IN-LS method (Aravkin et al., 2016) presented in Algorithm 1 which increases r to f^* and AM-FLS method presented in Algorithm 2 which decreases r to f^*.[3] We defer the illustration of Algorithm 2 to Figure 3 in Appendix A. The proposition below characterize how the complexity of Algorithm 1 and 2 depend on the complexity of the oracle.

Theorem 1. *The following statements hold:*
(a) *Algorithm 1 returns an ε-optimal and ε-feasible solution.*
(b) *Algorithm 2 returns an ε-optimal and feasible solution. Moreover, the solution $\mathbf{x}^{(k)}$ generated in any iteration of Algorithm 2 is feasible.*
(c) *The total complexity of Algorithm 1 is at most*

$$\mathcal{C}(\varepsilon)\max\left\{\log_{\frac{2}{\theta}}\left(\frac{4\max\{|S(r^{(0)})||f^*-r^{(0)}|, L(r^{(0)})\}}{\theta\varepsilon}\right), 2\right\}$$

and the total complexity of Algorithm 2 is at most

$$\mathcal{C}\left(\frac{\beta|H(r^{(0)})|}{2\theta(1+(r^{(0)}-f^*)/\varepsilon)}\right)\frac{2\theta}{\beta}\log\left(\frac{2\theta}{\beta}\max\{\frac{r^{(0)}-f^*}{\varepsilon}, 1\}\right)$$

where $\beta := -\frac{H(r^{(0)})}{r^{(0)}-f^*} \in (0, 1]$.

In both complexities above, the factor involving $\mathcal{C}$ is from solving the subproblem (5) by oracle $\mathcal{A}$ and the logarithmic factor is mainly from searching for the level parameter.

3. Oracles for Level-Set Methods

In this section, we consider the optimization methods for (5) that can be used as an affine-minorant oracle in Algorithm 1 and 2. Whether an optimization method for (5) is a good candidate depends on the following two aspects.

Algorithm 1 IN-LS Method (Aravkin et al., 2016)

```
Input: $r^{(0)} < f^*$, $\varepsilon > 0$ and $\theta \in (1, 2)$
for $k = 0, 1, \ldots,$ do
    $(L(r^{(k)}), U(r^{(k)}), S(r^{(k)}), \mathbf{x}^{(k)}) = \mathcal{A}(r^{(k)}, \theta, \varepsilon)$
    if $U(r^{(k)}) \leq \varepsilon$ then
        Return $\mathbf{x}^{(k)}$
    else
        $r^{(k+1)} \leftarrow r^{(k)} - L(r^{(k)})/S(r^{(k)})$
    end if
end for
```

Algorithm 2 AM-FLS Method

```
Input: $r^{(0)} > f^*$, $\varepsilon > 0$ and $\theta \in (1, \infty)$
for $k = 0, 1, \ldots,$ do
    $(L(r^{(k)}), U(r^{(k)}), S(r^{(k)}), \mathbf{x}^{(k)}) = \mathcal{A}(r^{(k)}, \theta, \varepsilon)$
    if $L(r^{(k)}) \geq \varepsilon S(r^{(k)})$ then
        Return $\mathbf{x}^{(k)}$
    else
        $r^{(k+1)} \leftarrow r^{(k)} + U(r^{(k)})/2$
    end if
end for
```

Capability of generating $L(r)$ and $S(r)$. Although $U(r)$ can be easily obtained from any $\mathbf{x} \in \mathcal{X}$ as $U(r) = \mathcal{P}(r; \mathbf{x})$, the lower bound $L(r)$ and the slope $S(r)$ are not directly available from most primal optimization methods for (5).

Complexity for large-scale problems. The complexity of most first-order methods is the product of per-iteration cost and the number of iterations to ensure the outputs $(L(r), U(r), S(r))$. Deterministic methods must read the whole data at a cost of $O(nd)$ per-iteration, which can be prohibited for large-scale problems. On the other hand, a stochastic oracle based on sampling over data has a low per-iteration but potentially requires more iterations.

Next, we will discuss a few candidates for affine-minorant oracle and their potential issues from the two aspects above, which motivate our choice of oracle in this paper.

3.1. Challenges with Oracles based on Saddle-Point Formulation

Since $\mathcal{P}(r; \mathbf{x})$ is in general a non-smooth convex function regardless of the smoothness of f_i, one may use the standard subgradient method as the oracle which has a complexity of $\mathcal{C}(\varepsilon) = O(\frac{nd}{\varepsilon^2})$. Here, the factor $O(nd)$ is the cost of evaluating the subgradient of $\mathcal{P}(r; \mathbf{x})$, which requires reading through the whole data. If f_i is smooth for each i, the special maximization structure in $\mathcal{P}(r; \mathbf{x})$ allows using the smoothing technique (Beck & Teboulle, 2012) to construct a smooth approximation of $\mathcal{P}(r; \mathbf{x})$, which is then minimized by an accelerated gradient method. This approach will have

[3] Algorithm 1 increases r because $S(r^{(k)}) < 0$ while Algorithm 2 decreases r because $U(r^{(k)}) < 0$.

a complexity of $\mathcal{C}(\varepsilon) = O(\frac{nd}{\varepsilon})$ (Lin et al., 2017).

The factor $O(nd)$ in the complexity makes the methods above not scalable for large instances. Moreover, as we mentioned earlier, a lower bound $L(r)$ is not directly available from both methods. One possible solution is to set $L(r) = U(r) - \mathcal{G}$, where the upper bound $U(r) = \mathcal{P}(r; \bar{\mathbf{x}})$ for some $\bar{\mathbf{x}} \in \mathcal{X}$ and $\mathcal{G}$ is a computable quantity that satisfies $\mathcal{P}(r; \bar{\mathbf{x}}) - H(r) \leq \mathcal{G}$ (Lin et al., 2017). This inequality is available for an iterate $\bar{\mathbf{x}}$ in many first-order methods with $\mathcal{G}$ decreasing to zero as the iteration proceeds. However, $\mathcal{G}$ is the worst-case optimality gap, which can be used to derive the iteration complexity but will be too conservative to construct a tight $L(r)$. Besides, how to construct the slope $S(r)$ from a primal algorithm is still unknown.

In an alternative approach, we can reformulate subproblem (5) into an equivalent min-max saddle-point problem

$$H(r) = \min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\Delta} \sum_{i=0}^{m} y_i(f_i(\mathbf{x}) - r_i), \text{ where} \tag{6}$$

$\Delta := \{\mathbf{y} = (y_0, \ldots, y_m)^\top \in \mathbb{R}^{m+1} | \sum_{i=0}^m y_i = 1, y_i \geq 0\}$ and $r_0 = r$, and then solve (6) by a primal-dual optimization method. The potential solvers include the Mirror-Prox method (Nemirovski, 2004) for smooth f_i and the subgradient method for saddle-point problems (Nemirovski et al., 2009) for non-smooth f_i. These methods generate a pair of primal and dual solutions, denoted by $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$, with which we can construct $U(r) = \max_{\mathbf{y}\in\Delta} \sum_{i=0}^m y_i(f_i(\bar{\mathbf{x}}) - r_i)$ and $L(r) = \min_{\mathbf{x}\in\mathcal{X}} \sum_{i=0}^m \bar{y}_i(f_i(\mathbf{x}) - r_i)$. One can show that Property 5 in Definition 1 holds for this $L(r)$ with $S(r) = -\bar{y}_0$. However, the complexity of these methods is similar to the aforementioned primal methods in the factor $O(nd)$, and computing $L(r)$ requires solving a non-trivial minimization which makes this approach impractical.

3.2. The Proposed Solution: Saddle-Point Formulation by Persepective Function

To overcome the aforementioned issues in existing methods when used as oracles, we utilize the special structure of f_{ij} given in (3) and propose a new saddle-point formulation for (5). Based on convex conjugate, the function $H(r)$ can be reformulated as the value of a saddle-point problem

$$H(r) = \min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\Delta, \tilde{\boldsymbol{\alpha}}\in\mathbb{R}^n} \left\{ \begin{array}{l} \sum_{i=0}^m \frac{y_i \tilde{\boldsymbol{\alpha}}_i^\top \Theta_i \mathbf{x}}{n_i} \\ -\sum_{i=0}^m \sum_{j=1}^{n_i} \frac{y_i}{n_i} \phi_{ij}^*(\tilde{\alpha}_{ij}) - \mathbf{y}^\top \mathbf{r} \end{array} \right\}$$

where $\mathbf{r} = (r_0 = r, r_1, \ldots, r_m)^\top$, $\Theta_i := [\xi_{i1}, \ldots, \xi_{in_i}]^\top$ are the data matrices, and $\tilde{\boldsymbol{\alpha}} = (\tilde{\boldsymbol{\alpha}}_0^\top, \tilde{\boldsymbol{\alpha}}_1^\top, \ldots, \tilde{\boldsymbol{\alpha}}_m^\top)^\top \in \mathbb{R}^n$ with $n := \sum_{i=0}^m n_i$ and $\tilde{\boldsymbol{\alpha}}_i = (\tilde{\alpha}_{i1}, \ldots, \tilde{\alpha}_{in_i})^\top \in \mathbb{R}^{n_i}$ for $i = 0, 1, \ldots, m$ is the associated dual variable whose each coordinate corresponds to a data point in either the objective function or a constraint.

The objective function of this min-max problem is not jointly concave in $(\mathbf{y}, \tilde{\boldsymbol{\alpha}})$. However, with simple changes of variables, it can be reformulated as a convex-concave saddle-point problem. In particular, we define a new variable $\boldsymbol{\alpha}_i := y_i \tilde{\boldsymbol{\alpha}}_i$ for $i = 0, 1, \ldots, m$ such that $H(r)$ becomes

$$H(r) = \min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{w}\in\mathcal{W}} K(r; \mathbf{x}, \mathbf{w}) \tag{7}$$

where $\mathcal{W} := \Delta \times \mathbb{R}^n$, $\mathbf{w} = (\mathbf{y}, \boldsymbol{\alpha})$, and

$$\begin{aligned} K(r; \mathbf{x}, \mathbf{w}) &:= \sum_{i=0}^m \frac{\boldsymbol{\alpha}_i^\top \Theta_i \mathbf{x}}{n_i} - \sum_{i=0}^m \sum_{j=1}^{n_i} \frac{y_i}{n_i} \phi_{ij}^*(\frac{\alpha_{ij}}{y_i}) - \mathbf{y}^\top \mathbf{r} \\ &= \boldsymbol{\alpha}^\top A\mathbf{x} - \sum_{i=0}^m \sum_{j=1}^{n_i} \frac{y_i}{n_i} \phi_{ij}^*(\frac{\alpha_{ij}}{y_i}) - \mathbf{y}^\top \mathbf{r}. \end{aligned} \tag{8}$$

Here, $A := [\frac{\Theta_0^\top}{n_0}, \frac{\Theta_1^\top}{n_1}, \ldots, \frac{\Theta_m^\top}{n_m}]^\top$ is an $n \times d$ matrix formed by stacking the data matrices $\frac{\Theta_i}{n_i}$ vertically. The function $y\phi_{ij}^*(\frac{\alpha}{y}) : [0,1] \times \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is called the *perspective* of ϕ_{ij}^* and is jointly convex in (y, α) if ϕ_{ij}^* is convex. Strictly speaking, this perspective function equals $y\phi_{ij}^*(\frac{\alpha}{y})$ if $y > 0$ and equals zero if $y = 0$. We only use $y\phi_{ij}^*(\frac{\alpha}{y})$ for both cases under the convention that $0\phi_{ij}^*(\frac{\alpha}{0}) = 0$ for any α.

In the rest of the paper, the notation $\mathbf{w}$ and its versions with superscript and accent (e.g. $\mathbf{w}'$ and $\hat{\mathbf{w}}$) will always represent a vector like $(\mathbf{y}, \boldsymbol{\alpha})$ in $\mathcal{W}$ with the same superscript and accent on both components (e.g. $(\mathbf{y}', \boldsymbol{\alpha}')$ and $(\hat{\mathbf{y}}, \hat{\boldsymbol{\alpha}})$). Let

$$\mathcal{P}(r; \mathbf{x}) := \max_{\mathbf{w}\in\mathcal{W}} K(r; \mathbf{x}, \mathbf{w}), \mathcal{D}(r; \mathbf{w}) := \min_{\mathbf{x}\in\mathcal{X}} K(r; \mathbf{x}, \mathbf{w}).$$

Compared to (6), the new formulation (7) has an advantage that variables $\mathbf{x}$ and $\boldsymbol{\alpha}$ only interact in the bilinear term $\boldsymbol{\alpha}_i^\top \Theta_i \mathbf{x}$ such that both $\mathcal{P}(r; \mathbf{x})$ and $\mathcal{D}(r; \mathbf{w})$ can be evaluated easily for most commonly used loss function ϕ_{ij} and domain $\mathcal{X}$. As a result, for any solution $(\bar{\mathbf{x}}, \bar{\mathbf{w}}) \in \mathcal{X} \times \mathcal{W}$, we can construct $U(r)$ and $L(r)$ as $U(r) = \mathcal{P}(r; \bar{\mathbf{x}})$ and $L(r) = \mathcal{D}(r; \bar{\mathbf{w}})$.

Let $(\mathbf{x}^*, \mathbf{w}^* = (\mathbf{y}^*, \boldsymbol{\alpha}^*))$ be a saddle point of (7), namely,

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathcal{X}} K(r; \mathbf{x}, \mathbf{w}^*), \quad \mathbf{w}^* \in \arg\max_{\mathbf{w}\in\mathcal{W}} K(r; \mathbf{x}^*, \mathbf{w}).$$

With the bilinear structure in (8), we can use Mirror-Prox method or the primal-dual methods in Chambolle & Pock (2011) and Chambolle & Pock (2016) as an oracle to solve (7) at a complexity of $O(\frac{nd\|A\|_2}{\varepsilon})$, where $\|A\|_2$ is the operator norm of A. The factor $O(nd)$ here is from the matrix-vector multiplication $A\bar{\mathbf{x}}$ and $A^\top \bar{\boldsymbol{\alpha}}$ performed in each iteration of both methods. To reduce the per-iteration cost, one can utilize the finite-sum structure in $\boldsymbol{\alpha}^\top A\mathbf{x}$ to construct a stochastic gradient with reduced noise at a per-iteration cost of only $O(n + d)$. This technique is known as the stochastic variance-reduced gradient (SVRG) method which we will discuss in the next section.

3.3. SVRG Method for Saddle-Point Subproblem

The SVRG method was originally developed for finite-sum minimization with simple constraint (Johnson & Zhang, 2013; Defazio et al., 2014; Xiao & Zhang, 2014; Allen-Zhu & Yuan, 2016; Allen-Zhu, 2017). A primal-dual SVRG algorithm has been proposed by Palaniappan & Bach (2016) for finite-sum saddle-point problems under strong convexity assumption. However, some challenges arise in applying their methods to (7) and we will discuss these challenges and our solutions as follows.

Non-Strongly Convex: The SVRG method by Palaniappan & Bach (2016) require strong convexity in the saddle-point problem which is not available in (7). Therefore, we adapt the standard approach by solving the following strongly convex approximation to (7)

$$H_{\mu,\nu}(r) := \min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{w}\in\mathcal{W}} \left\{ K(r;\mathbf{x},\mathbf{w}) + \frac{\mu\|\mathbf{x}\|_2^2}{2} - \nu h_B(\mathbf{w}) \right\} \tag{9}$$

where $\mu > 0$, $\nu > 0$ and $h_B : \Delta \times \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is [4]

$$h_B(\mathbf{w}) := 2(1+B)^2 \left(\sum_{i=0}^{m} y_i \ln y_i + \ln m \right) + \sum_{i=0}^{m} \frac{\|\boldsymbol{\alpha}_i\|_2^2}{y_i} \tag{10}$$

where B is a positive constant to be determined later. Similar to $\mathcal{P}$ and $\mathcal{D}$, we define

$$\mathcal{P}_{\mu,\nu}(r;\mathbf{x}) := \max_{\mathbf{w}\in\mathcal{W}} K(r;\mathbf{x},\mathbf{w}) + \frac{\mu\|\mathbf{x}\|_2^2}{2} - \nu h_B(\mathbf{w})$$

$$\mathcal{D}_{\mu,\nu}(r;\mathbf{w}) := \min_{\mathbf{x}\in\mathcal{X}} K(r;\mathbf{x},\mathbf{w}) + \frac{\mu\|\mathbf{x}\|_2^2}{2} - \nu h_B(\mathbf{w}).$$

The strong convexity of the minimization problem in (9) is from the term $\|\mathbf{x}\|_2^2/2$. The strong convexity of the maximization problem in (9) is guaranteed by 1-strong convexity of h_B according to the following lemma.

Lemma 2 (Proposition 3.4, Hoda et al. (2010))**.** *For any $B > 0$ in (10), h_B is continuously differentiable and 1-strongly convex with respect to the norm $\|(\mathbf{y},\boldsymbol{\alpha})\|_{1,2} := \sqrt{\|\mathbf{y}\|_1^2 + \|\boldsymbol{\alpha}\|_2^2}$ on the bounded domain*

$$\mathcal{W}_B := \left\{ (\mathbf{y},\boldsymbol{\alpha}) \,\middle|\, \begin{array}{l} \mathbf{y} \in \text{int}\Delta, \boldsymbol{\alpha} = (\boldsymbol{\alpha}_i)_{i=0}^m, \boldsymbol{\alpha}_i = y_i\tilde{\boldsymbol{\alpha}}_i \\ \text{for some } \tilde{\boldsymbol{\alpha}}_i \in \mathbb{R}^{n_i} \textit{ with } \|\tilde{\boldsymbol{\alpha}}_i\|_2 \le B \end{array} \right\}.$$

The reason for us to introduce h_B instead of simply use the quadratic term $\|\mathbf{w}\|_2^2/2$ to gain strong convexity is that h_B allows a closed-form solution for the proximal mapping which is main step in each iteration of the SVRG method. We will discuss this property of h_B below.

Closed-form solution for proximal mapping: When applied to (7), the SVRG method based on Euclidean distance must solve a proximal mapping

$$\min_{\mathbf{w}\in\mathcal{W}} \frac{\|\mathbf{w}-\mathbf{w}'\|_2^2}{2\tau} + \sum_{i=0}^{m}\sum_{j=1}^{n_i} \frac{y_i}{n_i} \phi_{ij}^*\big(\frac{\alpha_{ij}}{y_i}\big) \tag{11}$$

for some $\mathbf{w}' = (\mathbf{y}',\boldsymbol{\alpha}')$ and $\tau > 0$ in each iteration in order to update $\mathbf{w}$ (Palaniappan & Bach, 2016). However, for most of the interesting loss function ϕ_{ij},this minimization problem does not have a closed-form solution. Same issue occurs if SVRG is applied to (9) with $h_B(\mathbf{w})$ replaced by $\|\mathbf{w}\|_2^2/2$. To address this issue, one key observation is that the function h_B not only provides strong convexity but also allows us to design a special Bregman divergence that can replace the Euclidean distance in (11) so that the proximal mapping can be solved in a closed-form. In particular, the Bregman divergence we consider is induced by h_B as

$$\begin{aligned} & D(\mathbf{w},\mathbf{w}') \\ := \; & h_B(\mathbf{w}) - h_B(\mathbf{w}') - \langle \nabla h_B(\mathbf{w}'), \mathbf{w}-\mathbf{w}' \rangle \\ = \; & 2(1+B)^2 \sum_{i=0}^{m} y_i \ln\left(\frac{y_i}{y_i'}\right) + \sum_{i=0}^{m} y_i \left\| \frac{\boldsymbol{\alpha}_i}{y_i} - \frac{\boldsymbol{\alpha}_i'}{y_i'} \right\|_2^2 . \end{aligned}$$

To simplify the notation, we define a function $G_\nu(\mathbf{w})$ as

$$G_\nu(\mathbf{w}) := \sum_{i=0}^{m}\sum_{j=1}^{n_i} \frac{y_i}{n_i} \phi_{ij}^*\big(\frac{\alpha_{ij}}{y_i}\big) + \mathbf{y}^\top \mathbf{r} + \nu h_B(\mathbf{w}).$$

The proximal mapping in the SVRG method based on $D(\mathbf{w},\mathbf{w}')$ can be formulated as

$$\min_{\mathbf{w}\in\mathcal{W}} -\boldsymbol{\alpha}^\top \mathbf{v} + G_\nu(\mathbf{w}) + \frac{D(\mathbf{w},\mathbf{w}')}{\tau} \tag{12}$$

for some $\mathbf{v}$, $\mathbf{w}' = (\mathbf{y}',\boldsymbol{\alpha}') \in \mathcal{W}_B$ and $\tau > 0$. The solution for (12) is characterized below. Its proof and the intuition behind the solution are postponed to Appendix D.

Proposition 1. *Given any $\mathbf{v}_i \in \mathbb{R}^{n_i}$, $\mathbf{w}' = (\mathbf{y}',\boldsymbol{\alpha}') \in \mathcal{W}_B$ and $\tau > 0$, let $\tilde{\boldsymbol{\alpha}}_i' := \frac{\boldsymbol{\alpha}_i'}{\mathbf{y}_i'}$ and $\tilde{\boldsymbol{\alpha}}_i^\#$ be the optimal solution of the following minimization problem*

$$\rho_i := \min_{\tilde{\boldsymbol{\alpha}}_i\in\mathbb{R}^{n_i}} \left\{ \begin{array}{l} -\tilde{\boldsymbol{\alpha}}_i^\top \mathbf{v}_i + \sum_{j=1}^{n_i} \frac{1}{n_i}\phi_{ij}^*(\tilde{\alpha}_{ij}) \\ +\nu\left\|\tilde{\boldsymbol{\alpha}}_i\right\|_2^2 + \frac{1}{\tau}\left\|\tilde{\boldsymbol{\alpha}}_i - \tilde{\boldsymbol{\alpha}}_i'\right\|_2^2 \end{array} \right\} \tag{13}$$

for $i = 0,1,\ldots,m$. Let $\boldsymbol{\rho} = (\rho_0,\rho_1,\ldots,\rho_m)^\top$. Then, $(\mathbf{y}^\#,\boldsymbol{\alpha}^\#) \in \mathcal{W}$ defined as follows is a solution to (12):

$$\begin{aligned} y_i^\# \;&:=\; \frac{(y_i')^{\frac{1}{\tau\nu+1}} \exp\left(-\frac{r_i+\rho_i}{2(1+B)^2(\nu+1/\tau)}\right)}{\sum_{l=0}^{m} \left\{ (y_l')^{\frac{1}{\tau\nu+1}} \exp\left(-\frac{r_l+\rho_l}{2(1+B)^2(\nu+1/\tau)}\right) \right\}} \\ \boldsymbol{\alpha}_i^\# \;&:=\; y_i^\#\tilde{\boldsymbol{\alpha}}_i^\# \quad \textit{for } i = 0,1,\ldots,m. \end{aligned}$$

[4] We define $\frac{\|\boldsymbol{\alpha}_i\|_2^2}{y_i}$ as $+\infty$ if $y_i = 0$ but $\boldsymbol{\alpha}_i \neq \mathbf{0}$ and as 0 if $y_i = 0$ and $\boldsymbol{\alpha}_i = \mathbf{0}$.

Algorithm 3 CheckGap$(\mathbf{x}, \mathbf{w}, \varepsilon, \theta)$

```
if $\left\{\begin{array}{l} 0 \leq \mathcal{D}(r;\mathbf{w}) \leq \mathcal{P}(r;\mathbf{x}) \leq \varepsilon \text{ or} \\ 0 \leq \mathcal{P}(r;\mathbf{x}) \leq \theta\mathcal{D}(r;\mathbf{w}) \text{ or} \\ \theta\mathcal{P}(r;\mathbf{x}) \leq \mathcal{D}(r;\mathbf{w}) < 0 \end{array}\right\}$ then
    Return "Succeed"
else
    Return "Continue"
end if
```

Note that, for many commonly used loss function ϕ_{ij}, the vector $\tilde{\boldsymbol{\alpha}}_i^{\#}$ can be solved at a cost of $O(n_i)$ in a closed form, which is the basis for many existing dual (Shalev-Shwartz & Zhang, 2013) or primal-dual algorithms (Zhang & Xiao, 2015) for unconstrained empirical risk minimization.

With these notations and preparations, the SVRG method applied to (9) is presented in Algorithm 4, where $A_{:k}$ and $A_{l:}$ denote the kth column and the lth row of A, respectively, and $\mathbf{x}_k$ and $\boldsymbol{\alpha}_l$ represent the kth coordinate of $\mathbf{x}$ and the lth coordinate of $\boldsymbol{\alpha}$, respectively. This algorithm is originally proposed by Palaniappan & Bach (2016) using Euclidean distance and generalized with Bregman divergence by Shi et al. (2017). Here, we use the special Bregman divergence D to facilitate the update on dual variables. Similar to the SVRG method for finite-sum minimization, Algorithm 4 runs in stages with each stage consisting of T inner iterations. At the beginning of stage s, a deterministic gradient of the bilinear part, i.e., $(\bar{\mathbf{u}}^{(s)}, \bar{\mathbf{v}}^{(s)})$, is computed at a reference point $(\bar{\mathbf{x}}^{(s)}, \bar{\mathbf{w}}^{(s)})$ using the full matrix A. Then, by sampling the rows and columns of A, a stochastic gradient $(\mathbf{v}^{(t)}, \mathbf{u}^{(t)})$ is constructed in inner iteration t to update the solution $(\mathbf{x}^{(t)}, \mathbf{w}^{(t)})$. The variance of this stochastic gradient will decrease to zero as $(\bar{\mathbf{x}}^{(s)}, \bar{\mathbf{w}}^{(s)})$ and $(\mathbf{x}^{(t)}, \mathbf{w}^{(t)})$ both approach the optimality. The reference point is updated only once per stage so that the complexity spent in computing the deterministic gradients remains low.

The per-iteration cost of the SVRG method is mainly from solving the two minimizations (proximal mappings). The first one can be solved in a closed form when $\mathcal{X}$ has a simple structure, e.g., a ℓ_1-ball or ℓ_2-ball. The second one can also be solved as described in Proposition 1. A subroutine given in Algorithm 3 is used to terminate Algorithm 4. Obviously, if Algorithm 3 returns "Succeed", the output constructed as

$$\begin{aligned} & (U(r), L(r), S(r), \bar{\mathbf{x}}) \\ = \quad & (\mathcal{P}(r;\bar{\mathbf{x}}^{(s)}), \mathcal{D}(r;\bar{\mathbf{w}}^{(s)}), -\bar{y}_0^{(s)}, \bar{\mathbf{x}}^{(s)}) \end{aligned}$$

will satisfy the properties in Definition 1.

The convergence of Algorithm 4 depends on the strong convexity of h_B which exists on $\mathcal{W}_B$. The following lemma (with proof in Appendix B) affirms that $\bar{\mathbf{w}}^{(s)}$ and $\mathbf{w}^{(t)}$ will stay in $\mathcal{W}_B$ for a particular $B > 0$ so that, by choosing that B in h_B, the strong convexity can be guaranteed.

Algorithm 4 SVRG$(\bar{\mathbf{x}}^{(0)}, \bar{\mathbf{w}}^{(0)}, \mu, \nu, \zeta, \varepsilon, \theta)$ for (9)

```
for $s = 0, 1, \ldots,$ do
    if $\left\{\begin{array}{l} \text{CheckGap}(\bar{\mathbf{x}}^{(s)}, \bar{\mathbf{w}}^{(s)}, \varepsilon, \theta) = \text{“Success” or} \\ \mathcal{P}_{\mu,\nu}(r;\bar{\mathbf{x}}^{(s)}) - \mathcal{D}_{\mu,\nu}(r;\bar{\mathbf{w}}^{(s)}) \leq \zeta \end{array}\right\}$
    then
        Return $(\bar{\mathbf{x}}^{(s)}, \bar{\mathbf{w}}^{(s)})$
    end if
    $(\mathbf{x}^{(0)}, \mathbf{w}^{(0)}) = (\bar{\mathbf{x}}^{(s)}, \bar{\mathbf{w}}^{(s)})$
    $(\bar{\mathbf{u}}^{(s)}, \bar{\mathbf{v}}^{(s)}) = (A^\top\bar{\boldsymbol{\alpha}}^{(s)}, A\bar{\mathbf{x}}^{(s)})$
    for $t = 0, 1, \ldots, T-1$ do
        Uniformly sample $k$ from $[d]$ and $l$ from $[n]$
        $\mathbf{v}^{(t)} = \bar{\mathbf{v}}^{(s)} + dA_{:k}\mathbf{x}_k^{(t)} - dA_{:k}\bar{\mathbf{x}}_k^{(s)}$
        $\mathbf{u}^{(t)} = \bar{\mathbf{u}}^{(s)} + nA_{l:}^\top\boldsymbol{\alpha}_l^{(t)} - nA_{l:}^\top\bar{\boldsymbol{\alpha}}_l^{(s)}$
        $\mathbf{x}^{(t+1)} = \arg\min_{\mathbf{x}\in\mathcal{X}} \mathbf{x}^\top\mathbf{u}^{(t)} + \frac{\mu\|\mathbf{x}\|_2^2}{2} + \frac{\|\mathbf{x}-\mathbf{x}^{(t)}\|_2^2}{2\sigma}$
        $\mathbf{w}^{(t+1)} = \arg\min_{\mathbf{w}\in\mathcal{W}} -\boldsymbol{\alpha}^\top\mathbf{v}^{(t)} + G_\nu(\mathbf{w}) + \frac{D(\mathbf{w},\mathbf{w}^{(t)})}{\tau}$
    end for
    $(\bar{\mathbf{x}}^{(s+1)}, \bar{\mathbf{w}}^{(s+1)}) = (\mathbf{x}^{(T)}, \mathbf{w}^{(T)})$
end for
```

Lemma 3. *Let Θ_{ik} be the kth column of Θ_i, $B_{\mathbf{x}} := \max_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x}\|_2$, $\tilde{\boldsymbol{\alpha}}_i^* := \frac{\boldsymbol{\alpha}_i^*}{y_i^*}$ and B be a constant that satisfies*

$$B \geq \max\left\{2\left\|\tilde{\boldsymbol{\alpha}}_i^*\right\|_2, \frac{8d\max_k\|\Theta_{ik}\|_2 B_{\mathbf{x}}}{\gamma}, 2\left\|\frac{\bar{\boldsymbol{\alpha}}_i^{(0)}}{\bar{y}_i^{(0)}} - \tilde{\boldsymbol{\alpha}}_i^*\right\|_2\right\}$$

for $i = 0, 1, \ldots, m$ if Assumption 2 (a) holds and satisfies

$$B \geq \max_{\tilde{\alpha}_{ij}\in\text{dom}\phi_{ij}} \|\tilde{\boldsymbol{\alpha}}_i\|_2 \textit{ for } i = 0, 1, \ldots, m$$

if Assumption 2 (b) holds. Then $\bar{\mathbf{w}}^{(s)}, \mathbf{w}^{(t)} \in \mathcal{W}_B$ for all $s, t \geq 0$ in Algorithm 4 as long as $\bar{\mathbf{w}}^{(0)} \in \mathcal{W}_B$.

The definition of saddle-point ensures that $\tilde{\boldsymbol{\alpha}}_{ij}^* \in \partial\phi_{ij}(\xi_{ij}\top\mathbf{x}^*)$ so that it is not hard to compute such a constant B based on $B_{\mathbf{x}}$. With B defined in Lemma 3, h_B is 1-strongly convex on the region Algorithm 4 is active on.

The convergence of SVRG for saddle-point problem has been proved by Palaniappan & Bach (2016) and Shi et al. (2017) in terms of the Euclidean distance and Bregman divergence from the iterate to the saddle point. We present the convergence of Algorithm 4 in terms of the primal-dual objective gap below. The proof follows the idea from Yu et al. (2015) and is given in the Appendix E just for completeness.

Theorem 2. *Let $\kappa = \frac{2\|A\|_{\max}^2}{\mu\nu}$, $\sigma = \frac{1}{12\kappa\mu}$, $\tau = \frac{1}{12\kappa\nu}$ and $T = \left(\frac{4}{3} + 16\kappa\right)\log(\frac{18}{5})$ where $\|A\|_{\max} := \sqrt{\max\{d\max_k\|A_{:k}\|_2^2, n\max_l\|A_{l:}\|_2^2\}}$. Algorithm 4 guarantees*

$$\begin{aligned} & \mathbb{E}\left[\mathcal{P}_{\mu,\nu}(r;\bar{\mathbf{x}}^{(s)}) - \mathcal{D}_{\mu,\nu}(r;\bar{\mathbf{w}}^{(s)})\right] && (14) \\ \leq \quad & (1/2)^s\,(1+12\kappa)\left(\mathcal{P}_{\mu,\nu}(r;\bar{\mathbf{x}}^{(0)}) - \mathcal{D}_{\mu,\nu}(r;\bar{\mathbf{w}}^{(0)})\right) \end{aligned}$$

Algorithm 5 $(U(r), L(r), S(r), \bar{\mathbf{x}}) = \mathcal{A}(r, \varepsilon, \theta)$

```
Choose $\hat{\mathbf{x}}^{(0)} \in \mathcal{X}$ and $\hat{\mathbf{w}}^{(0)} \in \mathcal{W}_B$
Set $\zeta_0 := \mathcal{P}(r; \hat{\mathbf{x}}^{(0)}) - \mathcal{D}(r; \hat{\mathbf{w}}^{(0)})$
for $p = 0, 1, \dots,$ do
    if CheckGap$(\hat{\mathbf{x}}^{(p)}, \hat{\mathbf{w}}^{(p)}, \varepsilon, \theta)$ = “Success” then
        Return $(U(r), L(r), S(r), \bar{\mathbf{x}}) = (\mathcal{P}(r; \hat{\mathbf{x}}^{(p)}), \mathcal{D}(r; \hat{\mathbf{w}}^{(p)}), -\hat{y}_0^{(p)}, \hat{\mathbf{x}}^{(p)})$
    else
        $(\hat{\mathbf{x}}^{(p+1)}, \hat{\mathbf{w}}^{(p+1)}) = $ SVRG$(\hat{\mathbf{x}}^{(p)}, \hat{\mathbf{w}}^{(p)}, \frac{\zeta_0}{2^{p+3} Q_{\mathbf{x}}}, \frac{\zeta_0}{2^{p+3} Q_{\mathbf{w}}}, \frac{\zeta_0}{2^{p+2}}, \varepsilon, \theta)$
    end if
end for
```

Moreover, the number of outer iterations Algorithm 4 runs before termination, denoted by $\mathcal{S}$, satisfies

$$\mathbb{E}[\mathcal{S}] \leq 2 + 2 \log \left(\frac{(2 + 24\kappa) \left[\mathcal{P}_{\mu,\nu}(r; \bar{\mathbf{x}}^{(0)}) - \mathcal{D}_{\mu,\nu}(r; \bar{\mathbf{w}}^{(0)}) \right]}{\zeta} \right).$$

Since each inner iteration of Algorithm 4 has a complexity of $O(n + d)$ while computing $(\bar{\mathbf{u}}^{(s)}, \bar{\mathbf{v}}^{(s)})$ at outer iteration has a complexity of $O(nd)$, the total expected complexity of Algorithm 4 is $\tilde{O}((nd + (n + d)\kappa) \log(\frac{1}{\zeta}))$.

3.4. Overall Complexity of Level-Set Methods

Algorithm 4 is for the strongly convex approximation problem (9) with fixed μ and ν. To solve the orginal saddle-point problem (7), one needs to apply Algorithm 4 to (9) with sequentially reduced μ and ν so that (9) approximates (7) more and more precisely according to the following lemma.

Lemma 4. *For any $\mathbf{x} \in \mathcal{X}$ and $\mathbf{w} \in \mathcal{W}_B$, $|[\mathcal{P}(r; \mathbf{x}) - \mathcal{D}(r; \mathbf{w})] - [\mathcal{P}_{\mu,\nu}(r; \mathbf{x}) - \mathcal{D}_{\mu,\nu}(r; \mathbf{w})]| \leq \mu Q_{\mathbf{x}} + \nu Q_{\mathbf{w}}$ where $Q_{\mathbf{x}} := \max_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x}\|_2^2$ and $Q_{\mathbf{w}} := 2 \max_{\mathbf{w} \in \mathcal{W}_B} h_B(\mathbf{w})$ with B defined as in Lemma 3.*

The proof of this lemma is given in Appendix F. Then the algorithm for solving (7) is presented in Algorithm 5 with $Q_{\mathbf{x}}$ and $Q_{\mathbf{w}}$ defined as in Lemma 4, which will be used as the oracle $\mathcal{A}(r, \theta)$ in level-set methods.

Theorem 3. *Algorithm 5 is an affine-minorant oracle in Definition 1 with the expected complexity upper bound function $\mathcal{C}(z)$ satisfies $\mathcal{C}(z) = \tilde{O}\left(nd + (n + d)\frac{\|A\|_{\max}^2}{z^2}\right)$ as z decreases to zero, where $\|A\|_{\max}^2$ defined in Theorem 2.*

As a consequence of Theorem 1 and 3, if using Algorithm 5 as the oracle, Algorithm 1 returns an ε-optimal and ε-feasible solution with the complexity of $\tilde{O}(nd + \frac{n+d}{\varepsilon^2}\|A\|_{\max}^2)$ while Algorithm 2 returns an ε-optimal with a feasible solution path with similar complexity. If a deterministic saddle-point algorithm, e.g., Mirror-Prox method, is used as the oracle for solving (7), Algorithm 1 and 2 will have complexity of $\tilde{O}(\frac{nd}{\varepsilon}\|A\|_2)$. It is known that $\frac{1}{\max\{n,d\}}\|A\|_{\max}^2 \leq \|A\|_2^2 \leq \|A\|_{\max}^2$ so that the complexity by using Algorithm 5 as oracle can be lower for some regime of parameters, for example, when $n \geq d$ and $\varepsilon d \geq \sqrt{n}\|A\|_{\max}$. We acknowledge that there is potential to further reduce the complexity of the SVRG oracle and Algorithm 1 and 2 to $\tilde{O}\left(nd + (n + d)\frac{\|A\|_{\max}\sqrt{nd}}{\varepsilon}\right)$ by direct acceleration using auxiliary sequences or the catalyst technique (Lin et al., 2015; Palaniappan & Bach, 2016; Xiao et al., 2017). However, the SVRG methods accelerated in either way in literature all require Euclidean distance and the similar result for Bregman divergence does not exist. We leave it as a future work to accelerate our methods.

4. Discussion and Related Work

Lan & Zhou (2016) propose a stochastic gradient method for convex optimization with a single expectation constraint. Since a finite-sum constraint is a special case of an expectation constraint, the method by Lan & Zhou (2016) can be applied to (1) when $m = 1$. Yu et al. (2017) propose a different stochastic gradient method for stochastic and online optimization with multiple expectation constraints. However, both methods by Lan & Zhou (2016) and Yu et al. (2017) only ensures ε-feasibility after convergence while our AM-FLS can ensure a feasible solution path.

The IN-LS method presented in Algorithm 1 is proposed by Aravkin et al. (2016) without a detail discussion on the choice of oracles. In this paper, we propose an oracle based on the SVRG method with new Bregman divergence so that it can be applied to large-scale problem. The method presented in Algorithm 2 is an variant of the feasible level-set method by Lin et al. (2017). Compared to Lin et al. (2017), Algorithm 2 has a more efficient stopping criterion by using the slope $S(r)$ from the oracle while Lin et al. (2017) does not require $S(r)$ so that it can be applied with more general oracles. Additionally, the algorithm by Lin et al. (2017) only ensure a feasible and relative ε-optimal, namely, a solution feasible solution $\bar{\mathbf{x}}$ with $f(\bar{\mathbf{x}}) - f^* \leq \varepsilon(f(\mathbf{x}') - f^*)$, where $\mathbf{x}'$ is an initial strictly feasible solution. On the contrary, our AM-FLS can ensure a feasible and absolutely ε-optimal solution.

5. Numerical Results

In this section, we evaluate the numerical performance of the proposed methods on Neyman-Pearson classification problem (Tong et al., 2016) formulated as in (4), where we choose ϕ to be the smoothed hinge loss function, i.e., a function $\phi(z)$ that equals $\frac{1}{2} - z$, if $z \leq 0$, $\frac{1}{2}(1 - z)^2$ if $0 < z \leq 1$, and 0 if $z > 1$. We will compared the IN-LS method and our AM-FLS method using Algorithm 5 and the deterministic Mirror-Prox method (Nemirovski, 2004) to solve (6) as oracles. The comparison also involve the stochastic gradient methods by Lan & Zhou (2016) and Yu et al. (2017). The dataset we use is the rcv1 training data set from LIBSVM library [5]. It has $n = 20,242$ data points with a dimension of $d = 47,236$, among which the $n_0 = 10,491$ positive data points are used in the objective function and $n_1 = 9,751$ negative data points are used in the constraint of (4). We choose $\lambda = 5$ and $r_1 = 0.1$ in (4).

The numerical comparisons are conducted in two different scenarios. In the first scenario, all methods are initialized at

$$\mathbf{x}_{\text{min-obj}} := \mathop{\arg\min}_{\|\mathbf{x}\|_2 \le \lambda} \left\{ f_0(\mathbf{x}) = \frac{1}{n_0} \sum_{j=1}^{n_0} \phi(\mathbf{x}^\top \xi_{0j}) \right\}$$

which is super-optimal but infeasible (i.e., $f_1(\mathbf{x}_{\text{min-obj}}) > r_1$). In the second scenario, all algorithms are initialized at

$$\mathbf{x}_{\text{min-cst}} := \mathop{\arg\min}_{\|\mathbf{x}\|_2 \le \lambda} \left\{ f_1(\mathbf{x}) = \frac{1}{n_1} \sum_{j=1}^{n_1} \phi(-\mathbf{x}^\top \xi_{1j}) \right\}$$

which is strictly feasible to (4) by not optimal. The IN-LS method is applied in the first scenario where we choose the initial level $r^{(0)} = f_0(\mathbf{x}_{\text{min-obj}}) < f^*$ and $\theta = 1.2$. The AM-FLS method is applied in the second scenario where we choose the initial level $r^{(0)} = f_0(\mathbf{x}_{\text{min-cst}}) > f^*$ and $\theta = 5$. The precision level ε is set to be 10^{-8} in both level-set methods. The inner loop of SVRG is terminated after passing the data set twice. We choose τ and σ to be 10 and $\zeta_0 = 10^{-3}$ in SVRG instead of the theoretical values for a good practical performance. Hence, for a fair comparison, the step length parameters in the Mirror-Prox method and the methods by Lan & Zhou (2016) and Yu et al. (2017) are also tuned for good practical performances. All the stochastic methods in the comparisons are implemented using mini-batch to construct the (standard or variance-reduce) stochastic gradients with a batch size of 5000.

The numerical results in these two scenarios are presented in Figure 1 and 2, where the x-axis represents the number of data passes each algorithm performed while the y-axis represents the logarithm of a joint measurement of the optimality and the feasibility, namely, $\mathcal{P}(f^*; \mathbf{x}) = \max\{f_0(\mathbf{x}) - f^*, f_1(\mathbf{x}) - r_1\}$, which is zero only when the solution is both optimal and feasible. Here, the optimal value f^* is obtained by running the IN-LS method with Algorithm 5 as the oracle for a large number of iterations. In both scenarios, level-set methods with either oracle will be slower at the beginning (before 2000 data passes) than the stochastic gradient methods. However, the level-set methods will outperform as the number of data passes increase. Eventually, the level-set methods can reduce $\mathcal{P}(f^*; \mathbf{x})$ to a low level, which the stochastic gradient methods will need a prohibitively large number of iterations to achieve. By comparing the two oracles used in the same level-set method in both scenario, we conclude the our SVRG oracle with the special Bregman divergence is comparable with and sometimes more effective than the deterministic oracle. The jumps observed in the curves of both level-set methods happen at the moment when the level parameter $r^{(k)}$ is updated, which change the subproblem (7) so that the solution optimizing the previous subproblem has to be changed significantly.

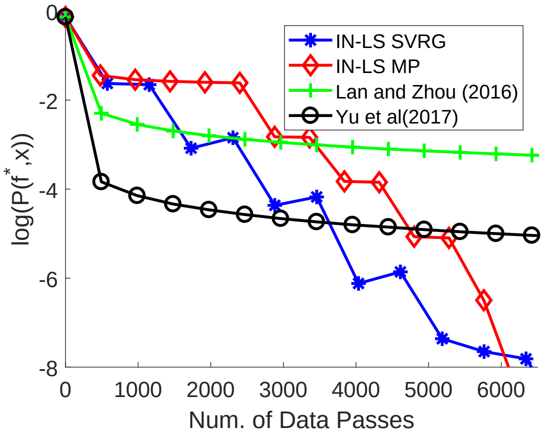

Figure 1. The convergence of $\mathcal{P}(f^*; \mathbf{x})$ to zero in each method when initialized at the super-optimal solution $\mathbf{x}_{\text{min-obj}}$.

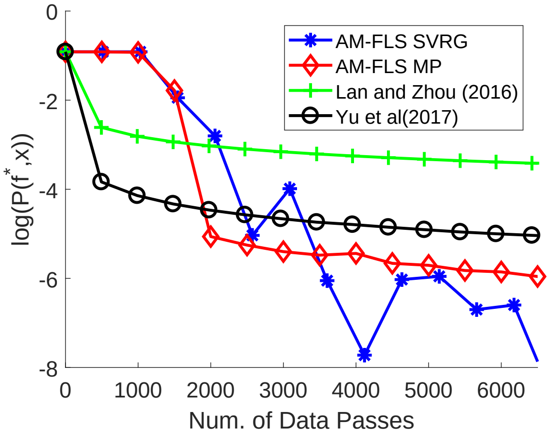

Figure 2. The convergence of $\mathcal{P}(f^*; \mathbf{x})$ to zero in each method when initialized at the strictly feasible solution $\mathbf{x}_{\text{min-cst}}$.

6. Conclusion

This paper introduces new numerical schemes for finite-sum constrained convex optimization. A new affine-minorized feasible level-set method is proposed which can guarantee a feasible solution path and absolutely ε-optimal solution. Moreover, we propose a new oracle for our level-set method based on the SVRG technique. This oracle leads to a lower total complexity of our level-set method.

[5] www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

Acknowledgements

The authors thank the reviewers for valuable feedback. Tianbao Yang was partially supported by NSF (IIS-1545995).

References

Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM Symposium on Theory of Computing*, STOC '17, 2017.

Allen-Zhu, Z. and Yuan, Y. Improved svrg for non-strongly-convex or sum-of-non-convex objectives. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1080–1089, 2016.

Aravkin, A. Y., Burke, J. V., Drusvyatskiy, D., Friedlander, M. P., and Roy, S. Level-set methods for convex optimization. *arXiv preprint arXiv:1602.01506*, 2016.

Barba-Gonzaléz, C., García-Nieto, J., Nebro, A. J., and Aldana-Montes, J. F. Multi-objective big data optimization with jmetal and spark. In *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 16–30. Springer, 2017.

Beck, A. and Teboulle, M. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

Bertsekas, D. P. *Nonlinear programming*. Athena scientific Belmont, 1999.

Bertsekas, D. P. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

Chambolle, A. and Pock, T. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

Chambolle, A. and Pock, T. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 159(1-2):253–287, 2016.

Cotter, A., Gupta, M., and Pfeifer, J. A light touch for heavily constrained sgd. In *Conference on Learning Theory*, pp. 729–771, 2016.

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

Fard, M. M., Canini, K., Cotter, A., Pfeifer, J., and Gupta, M. Fast and flexible monotonic functions with ensembles of lattices. In *Advances in Neural Information Processing Systems*, pp. 2919–2927, 2016.

Hoda, S., Gilpin, A., Pena, J., and Sandholm, T. Smoothing techniques for computing nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2):494–512, 2010.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pp. 315–323, 2013.

Lan, G. and Zhou, Z. Algorithms for stochastic optimization with expectation constraints. *arXiv preprint arXiv:1604.03887*, 2016.

Lemaréchal, C., Nemirovskii, A., and Nesterov, Y. New variants of bundle methods. *Mathematical programming*, 69(1):111–147, 1995.

Lim, E. On convergence rates of convex regression in multiple dimensions. *INFORMS Journal on Computing*, 26 (3):616–628, 2014.

Lin, H., Mairal, J., and Harchaoui, Z. A universal catalyst for first-order optimization. In *NIPS*, 2015.

Lin, Q., Nadarajah, S., and Sohelii, N. A level-set method for convex optimization with a feasible solution path. *Optimization Online*, 2017.

Mahdavi, M., Yang, T., and Jin, R. Stochastic convex optimization with multiple objectives. In *Advances in Neural Information Processing Systems*, pp. 1115–1123, 2013.

Marler, R. T. and Arora, J. S. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

Nemirovski, A. Prox-method with rate of convergence o(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15:229–251, 2004.

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

Nesterov, Y. *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied optimization*. Kluwer Academic Publishers, 2004.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

Palaniappan, B. and Bach, F. Stochastic variance reduction methods for saddle-point problems. In *Advances in Neural Information Processing Systems*, pp. 1408–1416, 2016.

Rigollet, P. and Tong, X. Neyman-pearson classification, convexity and stochastic constraints. *Journal of Machine Learning Research*, 12(Oct):2831–2855, 2011.

Ruszczyński, A. P. *Nonlinear optimization*, volume 13. Princeton university press, 2006.

Seijo, E., Sen, B., et al. Nonparametric least squares estimation of a multivariate convex regression function. *The Annals of Statistics*, 39(3):1633–1657, 2011.

Sen, B. and Meyer, M. Testing against a linear regression model using ideas from shape-restricted estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(2):423–448, 2017.

Shalev-Shwartz, S. and Zhang, T. Stochastic dual coordinate ascent methods for regularized loss minimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 567–599, 2013.

Shi, Z., Zhang, X., and Yu, Y. Bregman divergence for stochastic variance reduction: Saddle-point and adversarial prediction. In *Advances in Neural Information Processing Systems*, pp. 6033–6043, 2017.

Tong, X. A plug-in approach to neyman-pearson classification. *The Journal of Machine Learning Research*, 14(1): 3011–3040, 2013.

Tong, X., Feng, Y., and Zhao, A. A survey on neyman-pearson classification and suggestions for future research. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(2):64–81, 2016.

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

Xiao, L., Yu, A. W., Lin, Q., and Chen, W. Dscovr: Randomized primal-dual block coordinate algorithms for asynchronous distributed optimization. *arXiv preprint arXiv:1710.05080*, 2017.

Yu, A. W., Lin, Q., and Yang, T. Doubly stochastic primal-dual coordinate method for empirical risk minimization and bilinear saddle-point problem. *arXiv preprint arXiv:1508.03390*, 2015.

Yu, H., Neely, M., and Wei, X. Online convex optimization with stochastic constraints. In *Advances in Neural Information Processing Systems 30*, pp. 1427–1437. Curran Associates, Inc., 2017.

Zhang, Y. and Xiao, L. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *ICML*, pp. 353–361, 2015.

Zhao, A., Feng, Y., Wang, L., and Tong, X. Neyman-pearson classification under high-dimensional settings. *arXiv preprint arXiv:1508.03106*, 2015.