Some Worst-Case Datasets of Deterministic First-Order Methods for Solving Binary Logistic Regression

Yuyuan Ouyang   YUYUANO@CLEMSON.EDU
School of Mathematical and Statistical Sciences
Clemson University
Clemson, SC 29634, USA

Trevor Squires   TSQUIRE@CLEMSON.EDU
School of Mathematical and Statistical Sciences
Clemson University
Clemson, SC 29634, USA

Editor:

Abstract

We present in this paper some worst-case datasets of deterministic first-order methods for solving large-scale binary logistic regression problems. Under the assumption that the number of algorithm iterations is much smaller than the problem dimension, with our worst-case datasets it requires at least $\mathcal{O}(1/\sqrt{\varepsilon})$ first-order oracle inquiries to compute an ε-approximate solution. From traditional iteration complexity analysis point of view, the binary logistic regression loss functions with our worst-case datasets are new worst-case function instances among the class of smooth convex optimization problems.

Keywords: Binary Logistic Regression, Nonlinear Optimization, First-Order Methods, Lower Complexity Bound, Information-Based Complexity

1. Introduction

The following notations will be used throughout this paper. We denote natural logarithm by $\log(\cdot)$. For any positive integer k, we use $\mathbf{0}_k$ and $\mathbf{1}_k$ to denote k-dimensional vectors of all zeros and ones, respectively. When the dimension k is evident, we may remove the subscript and simply use $\mathbf{0}$ and $\mathbf{1}$. We use $\boldsymbol{e}_{t,k}$ to denote the t-th standard basis vector in $\mathbb{R}^k$: $\boldsymbol{e}_{t,k}^\top = (\mathbf{0}_{t-1}^\top, 1, \mathbf{0}_{k-t}^\top)^\top$. For any vector $\boldsymbol{u}$, we use $u^{(i)}$ to denote the i-th component of $\boldsymbol{u}$. The norm notation $\|\cdot\|$ is used for the Euclidean norm of a vector and the spectral norm of a matrix.

The main research questions of this paper are the following:

- *For any deterministic first-order methods, what is the best possible computational performance on solving large-scale binary logistic regression problems?*

- *For any deterministic first-order methods, what is their respective worst-case datasets of large-scale binary logistic regression problems that yield their worst possible computational performance?*

Note that we will focus on providing an answer to the second question, since it will lead natural to an answer to the first question. We describe the binary logistic regression prob-

lems, and provide the definitions of "deterministic first-order methods" and "computational performance" in the sequel.

In this paper, we use the following description of binary logistic problems. Given any data matrix $A \in \mathbb{R}^{N \times n}$ and response vector $\boldsymbol{b} \in \{-1, 1\}^N$, the binary logistic regression problem is a nonlinear optimization problem that minimizes objective function

$$\min_{\boldsymbol{x} \in \mathbb{R}^n, y \in \mathbb{R}} \Phi_{A,\boldsymbol{b}}(\boldsymbol{x}, y) := h(A\boldsymbol{x} + y\mathbf{1}) - \boldsymbol{b}^\top(A\boldsymbol{x} + y\mathbf{1}), \tag{1.1}$$

where for any $\boldsymbol{u} \in \mathbb{R}^k$, h is defined by

$$
\begin{aligned}
h(\boldsymbol{u}) \equiv h_k(\boldsymbol{u}) &:= \sum_{i=1}^k 2\log\left[2\cosh\left(\frac{u^{(i)}}{2}\right)\right] \\
&= \sum_{i=1}^k 2\log\left[\exp\left(\frac{u^{(i)}}{2}\right) + \exp\left(-\frac{u^{(i)}}{2}\right)\right].
\end{aligned}
\tag{1.2}
$$

Here cosh is the hyperbolic cosine function. For convenience we remove the subscript k in the definition of h and allow the variable vector of h to be of any dimension. Using $\boldsymbol{a}_i^\top$ to denote the i-th row of A, from (1.1) and (1.2) we have

$$
\begin{aligned}
\Phi_{A,\boldsymbol{b}}(\boldsymbol{x}, y) &= \sum_{i=1}^N 2\log\left[\exp\left(\frac{\boldsymbol{a}_i^\top \boldsymbol{x} + y}{2}\right) + \exp\left(-\frac{\boldsymbol{a}_i^\top \boldsymbol{x} + y}{2}\right)\right] - b^{(i)}\left(\boldsymbol{a}_i^\top \boldsymbol{x} + y\right) \\
&= \sum_{i=1}^N 2\log\left[\exp\left(\frac{b^{(i)}(\boldsymbol{a}_i^\top \boldsymbol{x} + y)}{2}\right) + \exp\left(-\frac{b^{(i)}(\boldsymbol{a}_i^\top \boldsymbol{x} + y)}{2}\right)\right] - b^{(i)}\left(\boldsymbol{a}_i^\top \boldsymbol{x} + y\right) \\
&= \sum_{i=1}^N 2\log\left[1 + \exp\left(-b^{(i)}(\boldsymbol{a}_i^\top \boldsymbol{x} + y)\right)\right],
\end{aligned}
$$

which is a commonly used form of binary logistic regression problems with parameter vector $\boldsymbol{x}$ and intercept y. Here in the second equality we use the fact that cosh is an even function and $b^{(i)} \in \{-1, 1\}$. Note that we can build an analogy between logistic and least squares problems through the formulation (1.1): if $h(\cdot) := \|\cdot\|^2/2$ we have a least squares problem immediately. In fact, such analogy has been exploited in Bach (2010) in the analysis of statistical properties of logistic regression.

In this paper, we will make an simplification and assume that we know the value of intercept y^* in an optimal solution $(\boldsymbol{x}^*, y^*)$. Problem (1.1) then simplifies to a problem of estimating the parameter vector $\boldsymbol{x}$ from

$$l_{A,\boldsymbol{b}}^* := \min_{\boldsymbol{x} \in \mathbb{R}^n} l_{A,\boldsymbol{b}}(\boldsymbol{x}) := \Phi_{A,\boldsymbol{b}}(\boldsymbol{x}, y^*).$$

Indeed, in our designed worst-case dataset, we can show that the intercept $y^* = 0$. In such case, it suffices to solve a logistic model with homogeneous linear predictor:

$$
\begin{aligned}
l_{A,b}^* := \min_{\boldsymbol{x} \in \mathbb{R}^n} l_{A,b}(\boldsymbol{x}) :=& h(A\boldsymbol{x}) - \boldsymbol{b}^\top A\boldsymbol{x} \\
=& \sum_{i=1}^N 2\log\left[1 + \exp\left(-b^{(i)}(\boldsymbol{a}_i^\top \boldsymbol{x})\right)\right].
\end{aligned}
\tag{1.3}
$$

The term "deterministic first-order method" is defined by the following oracle description: we say that an iterative algorithm $\mathcal{M}$ for convex optimization $\min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$ is a deterministic first-order method if it accesses the information of objective function f through a deterministic first-order *oracle* $\mathcal{O}_f : \mathbb{R} \times \mathbb{R}^n$, such that $\mathcal{O}_f(\boldsymbol{x}) = (f(\boldsymbol{x}), f'(\boldsymbol{x}))$ for any inquiry $\boldsymbol{x}$, where $f'(\boldsymbol{x})$ is a subgradient of f at $\boldsymbol{x}$. Specifically, $\mathcal{M}$ can be described by a problem independent initial iterate $\boldsymbol{x}_0$ and a sequence of rules $\{\mathcal{I}_t\}_{t=0}^\infty$ such that

$$\boldsymbol{x}_{t+1} = \mathcal{I}_t(\mathcal{O}_f(\boldsymbol{x}_0), \dots, \mathcal{O}_f(\boldsymbol{x}_t)), \ \forall t \geq 0. \tag{1.4}$$

Without loss of generality, we can assume that $\boldsymbol{x}_0 = \boldsymbol{0}$. We also assume that the dimension of the parameter vector $\boldsymbol{x}$ is large and we can only afford $T \ll n$ oracle inquiries.

The computational performance of $\mathcal{M}$ is evaluated through its solution accuracy $f(\hat{\boldsymbol{x}}) - f^*$ or $\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\|$, in which $\hat{\boldsymbol{x}}$ is an approximate solution computed by $\mathcal{M}$. Without loss of generality, we can assume that $\boldsymbol{x}_t$'s are both inquiry points to the oracle $\mathcal{O}$ and the approximate solution computed by $\mathcal{M}$.

1.1 Related Work

There had been many existing deterministic first-order algorithms that can be applied to solve (1.3). For example, applying Nesterov's accelerated gradient method (Nesterov, 2004), it is known that it takes at most $\mathcal{O}(1)(1/\sqrt{\varepsilon})$ oracle inquiries to compute an approximate solution $\hat{\boldsymbol{x}}$ to (1.3) such that $l_{A,\boldsymbol{b}}(\boldsymbol{x}) - l_{A,\boldsymbol{b}}^* \leq \varepsilon$. Here $\mathcal{O}(1)$ is a constant independent of ε. Such result is known as the *upper complexity bound.* Upper complexity bounds depict achievable computational performance on solving a specified class of problems.

Our research question described at the beginning Section 1 is focusing on the *lower complexity bound* of a problem, namely, the performance limit of deterministic first-order methods. For convex optimization problems $f^* := \min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$, the lower complexity bound is concerned with the least number of inquiries to the deterministic first-order oracle in order to compute an ε-approximate solution $\hat{\boldsymbol{x}}$ such that $f(\hat{\boldsymbol{x}}) - f^* \leq \varepsilon$. In the following we list the available lower complexity bound results on deterministic first-order methods for large-scale convex optimization $f^* := \min_{\boldsymbol{x} \in \mathbb{R}^n} f(\boldsymbol{x})$.

- When f is convex (possibly nonsmooth), the lower complexity bound is $\mathcal{O}(1)(1/\varepsilon^2)$ (Nemirovski and Yudin, 1983; Nesterov, 2004; Guzmán and Nemirovski, 2015; Woodworth and Srebro, 2016).

- When f is weakly smooth convex with parameter ρ (see the definition of weakly smooth functions in Juditsky and Nesterov, 2014), the lower complexity bound is $\mathcal{O}(1)(1/\varepsilon^{2/(1+3\rho)})$ (Juditsky and Nesterov, 2014; Guzmán and Nemirovski, 2015).

- When f is convex nonsmooth with bilinear saddle point structure, the lower complexity bound is $\mathcal{O}(1)(1/\varepsilon)$ (Ouyang and Xu, 2019).

- When f is smooth convex, the lower complexity bound is $\mathcal{O}(1)(1/\sqrt{\varepsilon})$ (Nemirovski, 1992; Nesterov, 2004; Guzmán and Nemirovski, 2015; Drori, 2017; Woodworth and Srebro, 2016; Carmon et al., 2017a,b; Diakonikolas and Guzmán, 2018).

- When f is strongly convex smooth, the lower complexity bound is $\mathcal{O}(1) \log(1/\varepsilon)$ (Nesterov, 2004; Woodworth and Srebro, 2016).

Two remarks are in place for the above list of lower complexity bounds. First, all the lower complexity bounds have been demonstrated to match available upper complexity bounds, namely, there exists available deterministic first-order algorithms that achieves the lower complexity bounds. Such algorithms are known as *optimal algorithms*, since the lower complexity bounds provide the verification that their respective theoretical computational performance would not be improvable anymore. Second, we can observe from the above list that the "smaller" the problem class is, the better lower complexity bounds could be. For example, the class of smooth convex optimization problems is a subclass of general convex optimization problems, hence it is possible to expect an algorithm with $\mathcal{O}(1/\sqrt{\varepsilon})$ upper complexity rather than $\mathcal{O}(1/\varepsilon^2)$.

1.2 Motivation

Our research question can be motivated by the second remark above, that is, a subclass might yield better lower complexity bounds. Note that the class of binary logistic regression problems is a subclass of the smooth convex problem class. Is it possible to design algorithms that targets solely on logistic regression, and performs better than the $\mathcal{O}(1)(1/\sqrt{\varepsilon})$ complexity bounds for smooth convex optimization? Unfortunately, such question has not yet been answered in the literature. Although there had been lower complexity bounds $\mathcal{O}(1)(1/\sqrt{\varepsilon})$ on smooth convex optimization (see Section 1.1), the worst-case instance functions provided for smooth convex optimization are either based on convex quadratic functions (Nemirovski, 1992; Nesterov, 2004; Drori, 2017; Woodworth and Srebro, 2016; Carmon et al., 2017a,b) or smoothing (through infimal convolution) of maximum of affine functions (Guzmán and Nemirovski, 2015; Diakonikolas and Guzmán, 2018). None of the provided worst-case instance is a binary logistic function.

The above discussion is based on the traditional perspective of complexity analysis of convex optimization, namely, finding worst-case functions among the problem class and explore the performance limits of algorithms. It is important to point out that our research question can also be viewed from one other perspective. In data analysis practice, we will usually designing algorithms that are tailored for specific models. Consequently, we are interested at exploring the performance limit of algorithms with respect to the *worst-case dataset*. From this perspective, our research question asks what the worst-case dataset that yields the worst performance of any deterministic first-order method. Note that the two aforementioned perspectives are equivalent; however, the latter one offers a more data-oriented argument.

In this paper, we describe some worst-case datasets of binary logistic regression problems, such that for any first-order methods, it requires at least $\mathcal{O}(1)(1/\sqrt{\varepsilon})$ first-order oracle inquiries to obtain an ε-approximate solution. Such datasets can be used as certificates of optimal deterministic first-order algorithms for binary logistic regression. Also, from the perspective of traditional complexity analysis, our results also provide new worst-case functions for smooth convex optimization.

In Section 2, we describe the construction of a worst-case dataset for deterministic first-order method that satisfy a mild assumption (see Assumption 2.1 below). In Section 3, we provide worst-case datasets for any given deterministic first-order method.

2. Worst-Case Dataset Under Linear Span Assumption

In this section, we make the following simple assumption regarding the iterates produced by a deterministic first-order method $\mathcal{M}$:

Assumption 2.1 *The iterate sequence $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots\}$ produced by $\mathcal{M}$ satisfies*

$$\boldsymbol{x}_t \in \operatorname{span}\{\nabla f(\boldsymbol{x}_0), \ldots, \nabla f(\boldsymbol{x}_{t-1})\}, \ \forall t \geq 1.$$

Recall that we have already made two assumptions in Section 1 (see the discussion after (1.4)) on $\mathcal{M}$ without loss of generality, namely, that $\boldsymbol{x}_0 = \boldsymbol{0}$ and that $\boldsymbol{x}_t$ is both the inquiry point and the output of approximate solution. By Assumption 2.1, the new iterate produced by $\mathcal{M}$ always lies inside the linear span of past gradients. Throughout this paper, we refer to Assumption 2.1 as the *linear span assumption*. Such linear span assumption, in the first look, does not seem to be one that can be made without loss of generality. However, we would like to emphasize here that the purpose of introducing the linear span assumption is only for us to demonstrate the lower complexity bound derivation in a straightforward manner; we will show in Section 3 that the linear span assumption can be removed, using the technique in the seminal work by Nemirovski (1992).

2.1 A Special Class of Datasets

We describe our construction of a special class of datasets for binary logistic regression. Such datasets will be used throughout this paper to construct worst-case datasets for binary logistic regression. Suppose that $\sigma > \zeta > 0$ are two fixed real numbers. Given any positive integer k, denote

$$W_k := \begin{pmatrix} & & & -1 & 1 \\ & & -1 & 1 & \\ & \ddots & \ddots & & \\ -1 & 1 & & & \\ 1 & & & & \end{pmatrix} \in \mathbb{R}^{k \times k} \tag{2.1}$$

and

$$A_k := \begin{pmatrix} 2\sigma W_k \\ -2\zeta W_k \\ -2\sigma W_k \\ 2\zeta W_k \end{pmatrix} \in \mathbb{R}^{4k \times k}, \ \boldsymbol{b}_k := \begin{pmatrix} \mathbf{1}_k \\ \mathbf{1}_k \\ -\mathbf{1}_k \\ -\mathbf{1}_k \end{pmatrix} \in \mathbb{R}^{4k}. \tag{2.2}$$

We then denote functions $f_k : \mathbb{R}^k \to \mathbb{R}$ and $\Phi_k : \mathbb{R}^{k+1} \to \mathbb{R}$ by

$$f_k(\boldsymbol{x}) := h(A_k \boldsymbol{x}) - \boldsymbol{b}_k^\top A_k \boldsymbol{x}_k \text{ and } \Phi_k(\boldsymbol{x}, y) := h(A_k \boldsymbol{x} + y\mathbf{1}_k) - \boldsymbol{b}_k^\top (A_k \boldsymbol{x}_k + y\mathbf{1}_k). \tag{2.3}$$

Comparing (2.3) with previous descriptions of $l_{A,b}$ and $\Phi_{A,b}$ in (1.1) and (1.3) respectively, f_k and Φ_k are clearly binary logistic regression objective functions with data matrix A_k and response vector $\boldsymbol{b}_k$: we are using logistic regression to train a classifier for two datasets

whose entries have opposite signs. Recall that $\sigma > \zeta > 0$; this assumption is to avoid duplicate data entries. Note that

$$
\begin{aligned}
\|A_k\boldsymbol{u}\|^2 =&8(\sigma^2 + \zeta^2)\|W_k\boldsymbol{u}\|^2 \\
=&8(\sigma^2 + \zeta^2)\left[\left(u^{(k)} - u^{(k-1)}\right)^2 + \ldots + \left(u^{(2)} - u^{(1)}\right)^2 + \left(u^{(1)}\right)^2\right] \\
\leq&16(\sigma^2 + \zeta^2)\left[\left(u^{(k)}\right)^2 + \left(u^{(k-1)}\right)^2 + \ldots + \left(u^{(2)}\right)^2 + \left(u^{(1)}\right)^2 + \left(u^{(1)}\right)^2\right] \\
\leq&32(\sigma^2 + \zeta^2)\|\boldsymbol{u}\|^2, \ \forall \boldsymbol{u} \in \mathbb{R}^k;
\end{aligned}
$$

consequently

$$
\|A_k\| \leq 4\sqrt{2(\sigma^2 + \zeta^2)}. \tag{2.4}
$$

In the following lemma, we describe the optimal solutions that minimizes f_k and Φ_k respectively. By the definition of f_k in (2.3) and noting the convexity of binary logistic regression problems, to solve a minimizer of f_k it suffices to solve

$$
\nabla f_k(\boldsymbol{x}) = A_k^\top \nabla h(A_k\boldsymbol{x}) - A_k^\top \boldsymbol{b}_k = 0. \tag{2.5}
$$

Noting the definition of h in (1.2), we have

$$
\nabla h(\boldsymbol{u}) = \tanh\left(\frac{\boldsymbol{u}}{2}\right) := \left(\tanh\left(\frac{u^{(1)}}{2}\right), \ldots, \tanh\left(\frac{u^{(k)}}{2}\right)\right)^\top, \ \forall \boldsymbol{u} \in \mathbb{R}^k, \ \forall k. \tag{2.6}
$$

Here tanh is the hyperbolic tangent function. Throughout this paper, we will slightly abuse the notation $\tanh(\boldsymbol{u})$ and allow the scalar function $\tanh(\cdot)$ to be applied to any vector $\boldsymbol{u}$ component-wisely.

Lemma 2.1 *For any $\sigma > \zeta > 0$, there always exists $c > 0$ that satisfies*

$$
\sigma \tanh(\sigma c) + \zeta \tanh(\zeta c) = \sigma - \zeta. \tag{2.7}
$$

Moreover,

$$
\boldsymbol{x}^* := c(1, 2, \ldots, k)^\top \tag{2.8}
$$

is the unique optimal solution to problem

$$
f_k^* := \min_{\boldsymbol{x} \in \mathbb{R}^k} f_k(\boldsymbol{x}) \tag{2.9}
$$

with

$$
f_k^* = 8k\log 2 + 4k\left\{\log\cosh(\sigma c) + \log\cosh(\zeta c) - (\sigma - \zeta)c\right\}. \tag{2.10}
$$

In addition, $(\boldsymbol{x}^*, 0)$ is the unique optimal solution to $\min_{\boldsymbol{x} \in \mathbb{R}^n, y \in \mathbb{R}} \Phi_k(\boldsymbol{x}, y)$.

Proof Note that there always exists $c > 0$ that satisfies (2.7) since the function $r(c) := \sigma \tanh(\sigma c) + \zeta \tanh(\zeta c) - \sigma + \zeta$ is continuous with $r(0) = -\sigma + \zeta < 0$ and $\lim_{c \to \infty} r(c) = 2\zeta > 0$.

By the definitions of W_k and $\boldsymbol{x}^*$ in (2.1) and (2.8), we observe that $W_k \boldsymbol{x}^* = c\mathbf{1}_k$ and $W_k \mathbf{1}_k = \boldsymbol{e}_{k,k}$. Using this observation and the descriptions of ∇h, A_k, and $\boldsymbol{b}_k$ in (2.6) and (2.2) respectively, and noting that tanh is an odd function and is assumed to apply to any vector component-wisely, we have

$$\nabla h(A_k \boldsymbol{x}^*) = \tanh \left[\frac{1}{2} \begin{pmatrix} 2\sigma c\mathbf{1}_k \\ -2\zeta c\mathbf{1}_k \\ -2\sigma c\mathbf{1}_k \\ 2\zeta c\mathbf{1}_k \end{pmatrix} \right] = \begin{pmatrix} \tanh(\sigma c)\mathbf{1}_k \\ -\tanh(\zeta c)\mathbf{1}_k \\ -\tanh(\sigma c)\mathbf{1}_k \\ \tanh(\zeta c)\mathbf{1}_k \end{pmatrix}, \tag{2.11}$$

$$A_k^\top \nabla h(A_k \boldsymbol{x}^*) = 4[\sigma \tanh(\sigma c) + \zeta \tanh(\zeta c)]\boldsymbol{e}_{k,k},$$
$$A_k^\top \boldsymbol{b}_k = 2(\sigma - \zeta + \sigma - \zeta)W_k \mathbf{1}_k = 4(\sigma - \zeta)\boldsymbol{e}_{k,k}. \tag{2.12}$$

Using the above results, the description of ∇f in (2.5), and the relation (2.7) that c satisfies, we have $\nabla f_k(\boldsymbol{x}^*) = 0$. Noting that binary logistic loss functions are strictly convex, we conclude that $\boldsymbol{x}^*$ is the unique minimizer of f_k. Recalling the definition of h in (1.2), noting that cosh is an even function, and using the computation of $A_k^\top \boldsymbol{b}_k$ in (2.12), we have

$$f_k^* = f_k(\boldsymbol{x}^*) = h(A_k \boldsymbol{x}^*) - (\boldsymbol{x}^*)^\top A_k^\top \boldsymbol{b}_k = h(A_k \boldsymbol{x}^*) - 4k(\sigma - \zeta)c$$
$$= 2k \{\log [2\cosh(\sigma c)] + \log [2\cosh(-\zeta c)] + \log [2\cosh(-\sigma c)] + \log [2\cosh(\zeta c)]\} - 4k(\sigma - \zeta)c$$
$$= 8k \log 2 + 4k \{\log \cosh(\sigma c) + \log \cosh(\zeta c) - (\sigma - \zeta)c\}.$$

Furthermore, by the descriptions of $\boldsymbol{b}_k$ and $\nabla h(A_k \boldsymbol{x}^*)$ in (2.2) and (2.11) respectively, computing the partial derivative of Φ in (2.3) with respective to y at 0, we have

$$\frac{\partial}{\partial y}\bigg|_{y=0} \Phi_k(\boldsymbol{x}^*, y) = \mathbf{1}_k^\top \nabla h(A_k \boldsymbol{x}^*) - \boldsymbol{b}_k^\top \mathbf{1}_k = 0.$$

Noting also that $\nabla_x \Phi_k(\boldsymbol{x}^*, 0) = \nabla f_k(\boldsymbol{x}^*) = 0$, we conclude that $(\boldsymbol{x}^*, 0)$ is the unique minimizer of the strictly convex binary logistic loss function $\Phi_k(\boldsymbol{x}, y)$. ∎

2.2 Lower Complexity Bound Under Linear Span Assumption

In this section, we study the lower complexity bound of deterministic first-order methods for solving the logistic regression problem (2.9), under the linear assumption described in Assumption 2.1.

Lemma 2.2 *Suppose that k and t are fixed positive integers such that $t \le k$. Define*

$$\mathcal{K}_{t,k} := \mathrm{span}\{\boldsymbol{e}_{k-t+1,k}, \ldots, \boldsymbol{e}_{k,k}\}, \ \forall k, \forall 1 \le t \le k. \tag{2.13}$$

Then for all $\boldsymbol{x} \in \mathcal{K}_{t,k}$, we have $A_k\boldsymbol{x}, \nabla h(A_k\boldsymbol{x}) \in \mathcal{J}_{t,k}$ and $A_k^\top \nabla h(A_k\boldsymbol{x}), \nabla f_k(\boldsymbol{x}) \in \mathcal{K}_{t+1,k}$, where

$$\mathcal{J}_{t,k} := \mathrm{span}\{\boldsymbol{e}_{1,4k}, \ldots, \boldsymbol{e}_{t,4k}, \boldsymbol{e}_{k+1,4k}, \ldots, \boldsymbol{e}_{k+t,4k}, \boldsymbol{e}_{2k+1,4k}, \ldots, \boldsymbol{e}_{2k+t,4k}, \boldsymbol{e}_{3k+1,4k}, \ldots, \boldsymbol{e}_{3k+t,4k}\}. \tag{2.14}$$

Moreover,

$$\min_{\boldsymbol{x} \in \mathcal{K}_{t,k}} f_k(\boldsymbol{x}) = 8(k-t)\log 2 + \min_{\boldsymbol{u} \in \mathbb{R}^t} f_t(\boldsymbol{u}). \tag{2.15}$$

Proof Fix $\boldsymbol{x} \in \mathcal{K}_{t,k}$. By (2.13) we have $\boldsymbol{x}^\top = (\boldsymbol{0}_{k-t}^\top, \boldsymbol{u}^\top)^\top$ for some $\boldsymbol{u} \in \mathbb{R}^t$. Thus by the definition of W_k in (2.1),

$$W_k\boldsymbol{x} = \left(\begin{array}{c|c} -1 & W_t \\ \hline W_{k-t} & \end{array} \right) \begin{pmatrix} \boldsymbol{0}_{k-t} \\ \boldsymbol{u} \end{pmatrix} = \begin{pmatrix} W_t\boldsymbol{u} \\ \boldsymbol{0}_{k-t} \end{pmatrix}.$$

Using the above result, the descriptions of A_k and ∇h in (2.2) and (2.6) respectively, and the definition of $\mathcal{J}_{t,k}$ in (2.14), we have

$$A_k\boldsymbol{x} = \begin{pmatrix} 2\sigma W_t\boldsymbol{u} \\ \boldsymbol{0}_{k-t} \\ -2\zeta W_t\boldsymbol{u} \\ \boldsymbol{0}_{k-t} \\ -2\sigma W_t\boldsymbol{u} \\ \boldsymbol{0}_{k-t} \\ 2\zeta W_t\boldsymbol{u} \\ \boldsymbol{0}_{k-t} \end{pmatrix} \in \mathcal{J}_{t,k}, \quad \nabla h(A_k\boldsymbol{x}) = \begin{pmatrix} \tanh(\sigma W_t\boldsymbol{u}) \\ \boldsymbol{0}_{k-t} \\ -\tanh(\zeta W_t\boldsymbol{u}) \\ \boldsymbol{0}_{k-t} \\ -\tanh(\sigma W_t\boldsymbol{u}) \\ \boldsymbol{0}_{k-t} \\ \tanh(\zeta W_t\boldsymbol{u}) \\ \boldsymbol{0}_{k-t} \end{pmatrix} \in \mathcal{J}_{t,k}. \tag{2.16}$$

Also, note that for all $\boldsymbol{v} \in \mathbb{R}^t$, the definition of W_k in (2.1) results in

$$W_k^\top \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{0}_{k-t} \end{pmatrix} = W_k \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{0}_{k-t} \end{pmatrix} = \left(\begin{array}{c|c} -1 & W_{k-t} \\ \hline W_t & \end{array} \right) \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{0}_{k-t} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0}_{k-t-1} \\ -\boldsymbol{v}^{(t)} \\ W_t\boldsymbol{v} \end{pmatrix} \in \mathcal{K}_{t+1,k} \tag{2.17}$$

Combining (2.16) and (2.17), and using the definition of A_k in (2.2) we have

$$A_k^\top \nabla h(A_k\boldsymbol{x}) = 2\sigma W_k^\top \begin{pmatrix} \tanh(\sigma W_t\boldsymbol{u}) \\ \boldsymbol{0}_{k-t} \end{pmatrix} - 2\zeta W_k^\top \begin{pmatrix} -\tanh(-\zeta W_t\boldsymbol{u}) \\ \boldsymbol{0}_{k-t} \end{pmatrix}$$

$$- 2\sigma W_k^\top \begin{pmatrix} -\tanh(\sigma W_t\boldsymbol{u}) \\ \boldsymbol{0}_{k-t} \end{pmatrix} + 2\zeta W_k^\top \begin{pmatrix} \tanh(\zeta W_t\boldsymbol{u}) \\ \boldsymbol{0}_{k-t} \end{pmatrix}$$

$$\in \mathcal{K}_{t+1,k}.$$

Using the above result and noting the value of $A_k^\top \boldsymbol{b}_k$ in (2.12), we conclude that

$$\nabla f_k(\boldsymbol{x}) = A_k^\top \nabla h(A_k\boldsymbol{x}) - A_k^\top \boldsymbol{b}_k \in \mathcal{K}_{t+1,k}.$$

To finish the proof it suffices to prove (2.15). By the definition of h in (1.2), the computations in (2.16), and the setting $\boldsymbol{x}^\top = (\boldsymbol{0}_{k-t}^\top, \boldsymbol{u}^\top)^\top$ we have

$$
\begin{aligned}
h(A_k \boldsymbol{x}) &= \sum_{i=1}^t 2\log(2\cosh(\sigma W_t u)^{(i)}) + (k-t) \cdot 2\log(2\cosh(0)) \\
&\quad + \sum_{i=1}^t 2\log(2\cosh(-\zeta W_t u)^{(i)}) + (k-t) \cdot 2\log(2\cosh(0)) \\
&\quad + \sum_{i=1}^t 2\log(2\cosh(-\sigma W_t u)^{(i)}) + (k-t) \cdot 2\log(2\cosh(0)) \\
&\quad + \sum_{i=1}^t 2\log(2\cosh(\zeta W_t u)^{(i)}) + (k-t) \cdot 2\log(2\cosh(0)) \\
&= 8(k-t)\log 2 + h(A_t \boldsymbol{u}).
\end{aligned}
$$

Also, noting that $\boldsymbol{x}^\top = (\boldsymbol{0}_{k-1}^\top, \boldsymbol{u}^\top)^\top$, by the description of $A_k^\top \boldsymbol{b}_k$ in (2.12) we have

$$
\boldsymbol{b}_k^\top A_k \boldsymbol{x} = 4(\sigma - \zeta)u^{(t)} = \boldsymbol{b}_t^\top A_t \boldsymbol{u}.
$$

Hence we conclude from the definition of $f_k(\boldsymbol{x})$ in (2.3) that $f_k(\boldsymbol{x}) = 8(k-t)\log 2 + f_t(\boldsymbol{u})$, and thus (2.15) holds. ∎

As an immediate consequence of the above lemma, in the following we show that the linear span assumption of a first-order method $\mathcal{M}$ will lead to $\boldsymbol{x}_t \in \mathcal{K}_{t,k}$ when minimizing $f_k(\boldsymbol{x})$.

Lemma 2.3 *Suppose that $\mathcal{M}$ is any deterministic first-order method that satisfies Assumption 2.1. When $\mathcal{M}$ is applied to minimize $f_k(\boldsymbol{x})$ in (2.3), we have $\boldsymbol{x}_t \in \mathcal{K}_{t,k}$ for all $1 \le t \le k$.*

Proof We prove the $t=1$ case first. By Assumption 2.1, $\boldsymbol{x}_1 \in \text{span}\{\nabla f_k(\boldsymbol{x}_0)\}$. Recalling the assumption that $\boldsymbol{x}_0 = \boldsymbol{0}$, we have $\nabla f_k(\boldsymbol{x}_0) = \nabla f_k(\boldsymbol{0}) = -A_k^\top \boldsymbol{b}_k$, and by the value of $A_k^\top \boldsymbol{b}_k$ in (2.12) we have $\nabla f_k(\boldsymbol{x}_0) \in \text{span}\{\boldsymbol{e}_{k,k}\}$. Noting the definition of $\mathcal{K}_{t,k}$ in (2.13) we have $\boldsymbol{x}_1 \in \mathcal{K}_{1,k}$.

Let us use induction and assume that $\boldsymbol{x}_i \in \mathcal{K}_{i,k}$ for all $1 \le i \le s < k$. By Lemma 2.2, we have $\nabla f_k(\boldsymbol{x}_i) \in \mathcal{K}_{i+1,k}$ for all s. Noting Assumption 2.1 we have

$$
\boldsymbol{x}_{s+1} \in \text{span}\{\nabla f_k(\boldsymbol{x}_0), \ldots, \nabla f_k(\boldsymbol{x}_s)\} \subseteq \mathcal{K}_{s+1,k}.
$$

Hence the induction is complete and we conclude that $\boldsymbol{x}_t \in \mathcal{K}_{t,k}$ for all $1 \le t \le k$. ∎

By the description of f_k^* in (2.10), the relation (2.15), and Lemma 2.3, we conclude that the error of iterate $\boldsymbol{x}_t$ in terms of objective function value can be lower bounded by

$$
\begin{aligned}
f_k(\boldsymbol{x}_t) - f_k^* &\ge \min_{\boldsymbol{x} \in \mathcal{K}_{t,k}} f_k(\boldsymbol{x}) - f_k^* = 8(k-t)\log 2 + f_t^* - f_k^* \\
&= 4(k-t)\left[(\sigma - \zeta)c - \log\cosh(\sigma c) - \log\cosh(\zeta c)\right].
\end{aligned}
\tag{2.18}
$$

In the following lemma, we provide a simplification of the above lower bound:

Lemma 2.4 *For any real numbers σ and ζ that satisfy $2\zeta > \sigma > \zeta > 0$, we have*

$$(\sigma - \zeta)c - \log\cosh(\sigma c) - \log\cosh(\zeta c) \geq c^2\sigma^2 C(\sigma/\zeta),$$

where $C(\sigma/\zeta)$ is a universal constant that depends only on the ratio σ/ζ. In particular, When $\sigma/\zeta = 1.3$, we have

$$C(1.3) > \frac{1}{2}. \tag{2.19}$$

Proof By checking its derivative it is easy to verify that the function $c \mapsto c\tanh(c)$ is increasing when $c > 0$. Hence we have

$$\zeta c\tanh(\zeta c) \leq \sigma c\tanh(\sigma c), \text{ i.e., } \zeta\tanh(\zeta c) \leq \sigma\tanh(\sigma c).$$

Applying the above relation to (2.7), we have

$$2\zeta\tanh(\zeta c) \leq \sigma - \zeta \leq 2\sigma\tanh(\sigma c).$$

Since tanh is an increasing function, we have from the above inequality that

$$c \in [c_{lb}, c_{ub}], \text{ where } c_{lb} := \frac{1}{\sigma}\operatorname{arctanh}\left(\frac{1}{2} - \frac{\zeta}{2\sigma}\right) \text{ and } c_{ub} := \frac{1}{\zeta}\operatorname{arctanh}\left(\frac{\sigma}{2\zeta} - \frac{1}{2}\right) \tag{2.20}$$

in which $c_{lb}, c_{ub} > 0$ are well-defined real numbers under the assumption that $2\zeta > \sigma > \zeta > 0$. Using the above result, the definition of c in (2.7), and noting that the function $c \mapsto c\tanh(c) - \log\cosh c$ is increasing when $c > 0$ (by checking its derivative), we have

$$\begin{aligned}
&(\sigma - \zeta)c - \log\cosh(\sigma c) - \log\cosh(\zeta c) \\
=&c^2\sigma^2 \frac{1}{c^2\sigma^2}\left[\sigma c\tanh(\sigma c) - \log\cosh(\sigma c) + \zeta c\tanh(\zeta c) - \log\cosh(\zeta c)\right] \\
\geq&c^2\sigma^2 C
\end{aligned}$$

where

$$C := \frac{1}{c_{ub}^2\sigma^2}\left[\sigma c_{lb}\tanh(\sigma c_{lb}) - \log\cosh(\sigma c_{lb}) + \zeta c_{lb}\tanh(\zeta c_{lb}) - \log\cosh(\zeta c_{lb})\right].$$

Noting (2.20), we can observe that the above constant C depends only on the ratio σ/ζ. The result (2.19) can then be computed numerically. ∎

We are now ready to state a lower complexity bound of deterministic first-order methods under the linear span assumption.

Theorem 2.5 *Suppose that $\mathcal{M}$ is any deterministic first-order method that satisfies the linear span assumption in Assumption 2.1. Given any iteration number T, there always exist data matrix $A \in \mathbb{R}^{N \times n}$ and response vector $\boldsymbol{b} \in \{-1, 1\}^N$, where $n = 2T$ and $N = 8T$,*

such that the T-th approximate solution $\boldsymbol{x}_T$ generated by $\mathcal{M}$ on minimizing the binary logistic loss function $l_{A,b}$ in (1.3) satisfies

$$
\begin{aligned}
l_{A,b}(\boldsymbol{x}_T) - l_{A,b}^* &> \frac{3\|A\|^2\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{32(2T+1)(4T+1)}, \\
\|\boldsymbol{x}_T - \boldsymbol{x}^*\|^2 &> \frac{1}{8}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2,
\end{aligned}
\tag{2.21}
$$

where x^* is the minimizer of f.

Proof Let us fix any $\zeta > 0$ and set $\sigma = 1.3\zeta$ in the definition of A_k in (2.2). By (2.4) we have

$$
\|A_k\| \leq 4\sqrt{2\sigma^2 + 2(\sigma/1.3)^2} < 8\sigma.
\tag{2.22}
$$

Let us apply $\mathcal{M}$ to minimize f_k defined in (2.3) where $k = 2T$. Recall that $\mathcal{M}$ starts at $\boldsymbol{x}_0 = 0$, and that the minimizer $\boldsymbol{x}^*$ in (2.8) satisfies

$$
\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2 = c^2 \sum_{i=1}^{k} i^2 = \frac{c^2}{6} k(k+1)(2k+1).
\tag{2.23}
$$

By Lemmas 2.3 and 2.4, the lower bound estimate (2.18), and noting that $\sigma > \zeta$, we have $\boldsymbol{x}_t \in \mathcal{K}_{t,k}$ and

$$
f_k(\boldsymbol{x}_t) - f_k^* \geq 2(k-t)c^2\sigma^2, \ \forall t \leq k.
$$

Applying (2.22) and (2.23), the above relation becomes

$$
f_k(\boldsymbol{x}_t) - f_k^* > \frac{3(k-t)\|A_k\|^2\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2}{16k(k+1)(2k+1)}.
\tag{2.24}
$$

Also, since $\boldsymbol{x}_t \in \mathcal{K}_{t,k}$, by the definition of $\mathcal{K}_t$ in (2.13) we have $x_t^{(1)} = \ldots = x_t^{(k-t)} = 0$. Noting the description of $\boldsymbol{x}^*$ in (2.8) and focusing on the difference between $\boldsymbol{x}_t$ and $\boldsymbol{x}^*$ in the first $(k-t)$ components, we have

$$
\|\boldsymbol{x}_t - \boldsymbol{x}^*\|^2 \geq c^2 \sum_{i=1}^{k-t} i^2 = \frac{c^2}{6}(k-t)(k-t+1)(2k-2t+1).
\tag{2.25}
$$

Specially, setting $t = T$ and recalling that $k = 2T$, (2.24) becomes

$$
f_k(\boldsymbol{x}_T) - f_k^* > \frac{3\|A_k\|^2\|x_0 - x^*\|^2}{32(2T+1)(4T+1)},
$$

and (2.23) and (2.25) imply that

$$
\|\boldsymbol{x}_T - \boldsymbol{x}^*\|^2 \geq \frac{c^2}{6}T(T+1)(2T+1) > \frac{c^2}{48} \cdot 2T(2T+1)(4T+1) = \frac{1}{8}\|\boldsymbol{x}_0 - \boldsymbol{x}^*\|^2.
$$

We conclude (2.21) from the above two results by setting $A := A_k \in \mathbb{R}^{8T \times 2T}$, $\boldsymbol{b} := \boldsymbol{b}_k \in \mathbb{R}^{8T}$ and noting the equivalence between $l_{A,b}$ in (1.3) and f_k in the above derivation. ∎

3. Lower Complexity Bound for General Deterministic First-Order Methods

In this section, we extend the lower complexity bound to general deterministic first-order methods. The derivation is based on the concept of orthogonal invariance in the seminal work of Nemirovski (1992), and is organized in a similar way as in Ouyang and Xu (2019). Note that we can also use the concept of zero-respecting algorithms in Carmon et al. (2017a,b) to finish the proof.

We will use the following technical lemma which is proved in Ouyang and Xu (2019) (see Lemma 3.1 within).

Lemma 3.1 *Let $\mathcal{X} \subsetneq \bar{\mathcal{X}} \subseteq \mathbb{R}^p$ be two linear subspaces. Then for any $\bar{\boldsymbol{x}} \in \mathbb{R}^p$, there exists an orthogonal matrix $V \in \mathbb{R}^{p \times p}$ such that*

$$V\boldsymbol{x} = \boldsymbol{x}, \ \forall \boldsymbol{x} \in \mathcal{X}, \ \text{and} \ V\bar{\boldsymbol{x}} \in \bar{\mathcal{X}}.$$

Proposition 3.2 *For any A_k and b_k in the form of (2.2), any deterministic first-order method $\mathcal{M}$, and any $t \leq (k-3)/2$, there exists an orthogonal matrix $U_t \in \mathbb{R}^{k \times k}$ that satisfy the following:*

1. *$U_t A_k^\top \boldsymbol{b}_k = A_k^\top \boldsymbol{b}_k$;*

2. *When $\mathcal{M}$ is applied to minimize the binary logistic regression loss function $l_{A_k U_t, \boldsymbol{b}_k}$ defined in (1.3), its iterates $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_t$ satisfy*

$$\boldsymbol{x}_i \in U_t^\top \mathcal{K}_{2i+1}, \ \forall i = 0, \ldots, t.$$

Proof Let us fix A_k, b_k and the method $\mathcal{M}$. Throughout this proof, we will use the notation

$$\mathcal{U} := \left\{ V \in \mathbb{R}^{k \times k} \ \middle| \ V \text{ is orthogonal and } V A_k^\top \boldsymbol{b}_k = A_k^\top \boldsymbol{b}_k \right\}.$$

We conduct the proof by induction. The case when $t = 0$ is trivial by setting U_0 to be the identity matrix. Let us assume that the proposition is true when $t = s - 1 < (k-1)/2$. By the induction hypothesis there exists $U_{s-1} \in \mathcal{U}$ such that when $\mathcal{M}$ is applied to minimize $l_{A_k U_{s-1}, \boldsymbol{b}_k}$, its iterates satisfy

$$\boldsymbol{x}_i \in U_{s-1}^\top \mathcal{K}_{2i+3,k}, \ \forall i = 0, \ldots, s - 1. \tag{3.1}$$

Suppose that $\boldsymbol{x}_s$ is the next iterate. To prove the case when $t = s$, let us start by finding an orthogonal matrix $U_s \in \mathcal{U}$. Noting that $s < (k-1)/2$, from the definition of $\mathcal{K}_{t,k}$ in (2.13) we have

$$\mathcal{K}_{1,k} \subsetneq \mathcal{K}_{2,k} \subsetneq \ldots \subsetneq \mathcal{K}_{2s+1,k}. \tag{3.2}$$

Thus $U_{s-1}^\top \mathcal{K}_{2s} \subsetneq U_{s-1}^\top \mathcal{K}_{2s+1}$, and by Lemma 3.1 there exists orthogonal matrix V such that

$$V\boldsymbol{x} = \boldsymbol{x}, \ \forall x \in U_{s-1}^\top \mathcal{K}_{2s}, \ \text{and} \ V\boldsymbol{x}_s \in U_{s-1}^\top \mathcal{K}_{2s+1}. \tag{3.3}$$

Let us define

$$U_s := U_{s-1}V. \tag{3.4}$$

Noting the descriptions of $A_k^\top \boldsymbol{b}_k$ and $\mathcal{K}_{1,k}$ in (2.12) and (2.13) respectively, we observe that $A_k^\top \boldsymbol{b}_k \in \mathcal{K}_{1,k} \subset \mathcal{K}_{2s,k}$. Using such observation, by (3.3), (3.4), and the induction hypothesis $U_{s-1} \in \mathcal{U}$, we have $U_s^\top A_k^\top \boldsymbol{b}_k = V^\top U_{s-1}^\top A_k^\top \boldsymbol{b}_k = A_k^\top \boldsymbol{b}_k$, hence $U_s \in \mathcal{U}$. Also, from (3.2) we have $U_{s-1}^\top \mathcal{K}_{2i+1,k} \subset U_{s-1}^\top \mathcal{K}_{2s}$ for all $i = 0, \ldots, s-1$. Consequently by (3.3) and (3.4) we have

$$U_s^\top \mathcal{K}_{2i+1,k} = V^\top U_{s-1}^\top \mathcal{K}_{2i+1,k} = U_{s-1}^\top \mathcal{K}_{2i+1,k}, \ \forall i = 1, \ldots, s-1. \tag{3.5}$$

Applying the above relation to (3.1) and also noting $\boldsymbol{x}_s \in U_s^\top \mathcal{K}_{2s+1}$ from (3.3) and (3.4), we obtain

$$\boldsymbol{x}_i \in U_s^\top \mathcal{K}_{2i+1,k}, \ \forall i = 0, \ldots, s. \tag{3.6}$$

Let us apply $\mathcal{M}$ to minimize $l_{A_k U_s, \boldsymbol{b}_k}$. We will prove that its first $s+1$ iterates are exactly $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_s$ (the ones computed when $\mathcal{M}$ is applied to $l_{A_k U_{s-1}, \boldsymbol{b}_k}$). Indeed, we can make the following observation: if

$$l_{A_k U_s, \boldsymbol{b}_k}(\boldsymbol{x}) = l_{A_k U_{s-1}, \boldsymbol{b}_k}(\boldsymbol{x}) \text{ and } \nabla l_{A_k U_s, \boldsymbol{b}_k}(\boldsymbol{x}) = \nabla l_{A_k U_{s-1}, \boldsymbol{b}_k}(\boldsymbol{x}), \ \forall \boldsymbol{x} \in U_s^\top \mathcal{K}_{2s-1,k}, \tag{3.7}$$

then by (3.6) and the oracle assumption (1.4), $\mathcal{M}$ would obtain exactly the same first-order information at $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{s-1} \in U_s^\top \mathcal{K}_{2s-1,k}$ from the first-order oracle when minimizing either $l_{A_k U_s, \boldsymbol{b}_k}$ or $l_{A_k U_{s-1}, \boldsymbol{b}_k}$. Therefore, if (3.7) holds, then $\mathcal{M}$ produces exactly the same iterates $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_s$ when minimizing either $l_{A_k U_s, \boldsymbol{b}_k}$ or $l_{A_k U_{s-1}, \boldsymbol{b}_k}$. Consequently, noting that $U_s \in \mathcal{U}$ and (3.6) we obtain the results of the $t = s$ case by choosing $U = U_s$ and complete the induction.

To finish the induction proof it suffices to prove (3.7). Let us fix any $\boldsymbol{x} \in U_s^\top \mathcal{K}_{2s-1,k}$. By (3.3) and (3.5) we have $\boldsymbol{x} \in U_{s-1}^\top \mathcal{K}_{2s-1,k}$. Noting (3.3) and that $U_{s-1}, U_s \in \mathcal{U}$, we obtain the following relations:

$$U_s \boldsymbol{x} = U_{s-1}V\boldsymbol{x} = U_{s-1}\boldsymbol{x} \in \mathcal{K}_{2s-1,k} \text{ and } U_s^\top A_k^\top \boldsymbol{b}_k = A_k^\top \boldsymbol{b}_k = U_{s-1}^\top A_k^\top \boldsymbol{b}_k. \tag{3.8}$$

Moreover, noting that $U_{s-1}\boldsymbol{x} \in \mathcal{K}_{2s-1,k}$, applying Lemma 2.2 we have $A_k^\top \nabla h(A_k U_{s-1}\boldsymbol{x}) \in \mathcal{K}_{2s,k}$, and hence by (3.3) we observe that $V^\top U_{s-1}^\top A_k^\top \nabla h(A_k U_{s-1}\boldsymbol{x}) = U_{s-1}^\top A_k^\top \nabla h(A_k U_{s-1}\boldsymbol{x})$. Using such observation, recalling the definition of $l_{A,\boldsymbol{b}}$ in (1.3), and noting the relations in (3.8), we conclude that

$$l_{A_k U_s, \boldsymbol{b}_k}(\boldsymbol{x}) = h(A_k U_s \boldsymbol{x}) - \boldsymbol{x}^\top U_s^\top A_k^\top \boldsymbol{b}_k = h(A_k U_{s-1}\boldsymbol{x}) - \boldsymbol{x}^\top U_{s-1}^\top A_k^\top \boldsymbol{b}_k = l_{A_k U_{s-1}, \boldsymbol{b}_k}(\boldsymbol{x}),$$
$$\nabla l_{A_k U_s, \boldsymbol{b}_k}(\boldsymbol{x}) = U_s^\top A_k^\top \nabla h(A_k U_s \boldsymbol{x}) - U_s^\top A_k^\top \boldsymbol{b}_k = V^\top U_{s-1}^\top A_k^\top \nabla h(A_k U_{s-1}\boldsymbol{x}) - U_{s-1}^\top A_k^\top \boldsymbol{b}_k$$
$$= U_{s-1}^\top A_k^\top \nabla h(A_k U_{s-1}\boldsymbol{x}) - U_{s-1}^\top A_k^\top \boldsymbol{b}_k = \nabla l_{A_k U_{s-1}, \boldsymbol{b}_k}(\boldsymbol{x}).$$

Hence (3.7) is proved.

■

Theorem 3.3 *Suppose that $\mathcal{M}$ is any deterministic first-order method. Given any iteration number T, there always exists data matrix $A \in \mathbb{R}^{N \times n}$ and $b \in \mathbb{R}^N$, where $n = 4T + 2$ and $N = 16T + 8$, such that the T-th approximate solution $\boldsymbol{x}_T$ generated by $\mathcal{M}$ on minimizing the binary logistic regression loss function $l_{A,b}$ in (1.3) satisfies*

$$l_{A,\boldsymbol{b}}(\boldsymbol{x}_T) - l_{A,\boldsymbol{b}}^* \leq \frac{3\|A\|^2 \|\boldsymbol{x}_0 - \boldsymbol{z}^*\|^2}{32(4T + 3)(8T + 5)}$$

$$\|\boldsymbol{x}_T - \boldsymbol{z}^*\|^2 > \frac{1}{8}\|\boldsymbol{x}_0 - \boldsymbol{z}^*\|^2,$$

where $\boldsymbol{z}^*$ is the minimizer of $l_{A,b}$.

Proof Let us fix any $\zeta > 0$ and set $\sigma = 1.3\zeta$ in the definition of A_k in (2.2), in which we set $k = 4T + 2$. Note that the norm of A_k satisfies (2.22). Applying Proposition 3.2 to A_k, b_k, and $\mathcal{M}$ with $t = T$, we obtain the following result: there exists an orthogonal matrix $U := U_T$ that satisfies $U^\top A_k^\top \boldsymbol{b}_k = A_k^\top \boldsymbol{b}_k$, such that when $\mathcal{M}$ is applied to minimize $l_{A_k U, \boldsymbol{b}_k}$, its iterates $\boldsymbol{x}_i$ satisfies $\boldsymbol{x}_i \in U^\top \mathcal{K}_{2i+1,k}$ for all $0 \leq i \leq T$. Note that in this result we have

$$
\begin{aligned}
l_{A_k U, \boldsymbol{b}_k}(\boldsymbol{x}_T) &\geq \min_{\boldsymbol{x} \in U^\top \mathcal{K}_{2T+1,k}} l_{A_k U, \boldsymbol{b}_k}(\boldsymbol{x}) = \min_{\boldsymbol{x} \in U^\top \mathcal{K}_{2T+1,k}} h(A_k U \boldsymbol{x}) - \boldsymbol{x}^\top U^\top A_k^\top \boldsymbol{b}_k \\
&= \min_{\boldsymbol{x} \in \mathcal{K}_{2T+1,k}} h(A_k \boldsymbol{x}) - \boldsymbol{x}^\top A_k^\top \boldsymbol{b}_k = \min_{\boldsymbol{x} \in \mathcal{K}_{2T+1,k}} f_k(\boldsymbol{x}) \text{ and} \\
l_{A_k U, \boldsymbol{b}_k}^* &= \min_{\boldsymbol{x} \in \mathbb{R}^k} l_{A_k U, \boldsymbol{b}_k}(\boldsymbol{x}) = \min_{\boldsymbol{x} \in \mathbb{R}^k} h(A_k U \boldsymbol{x}) - \boldsymbol{x}^\top U^\top A_k^\top \boldsymbol{b}_k \qquad (3.9) \\
&= \min_{\boldsymbol{x} \in \mathbb{R}^k} h(A_k \boldsymbol{x}) - \boldsymbol{x}^\top A_k^\top \boldsymbol{b}_k = \min_{\boldsymbol{x} \in \mathbb{R}^k} f_k(\boldsymbol{x}) = f_k^*.
\end{aligned}
$$

Here we use the definition of f_k in (2.3). Consequently,

$$l_{A_k U, \boldsymbol{b}_k}(\boldsymbol{x}_T) - l_{A_k U, \boldsymbol{b}_k}^* \geq \min_{\boldsymbol{x} \in \mathcal{K}_{2T+1,k}} f_k(\boldsymbol{x}) - f_k^*. \qquad (3.10)$$

Note from (3.9) above that the minimizer $\boldsymbol{z}^*$ of $l_{A_k U, \boldsymbol{b}_k}(\boldsymbol{x})$ satisfies $\boldsymbol{z}^* = U^\top \boldsymbol{x}^*$, where x^* is the minimizer of f_k defined in (2.8). Since $\boldsymbol{x}_T \in U^\top \mathcal{K}_{2T+1,k}$, we have

$$
\begin{aligned}
\|\boldsymbol{x}_T - \boldsymbol{z}^*\|^2 &\geq \max_{\boldsymbol{x} \in \mathcal{K}_{2T+1}} \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 \\
&\geq c^2 \sum_{i=1}^{k-2T-1} i^2 = \frac{c^2}{6}(k - 2T - 1)(k - 2T)(2k - 4T - 1) \\
&= \frac{c^2}{6}(2T + 1)(2T + 2)(4T + 1).
\end{aligned}
$$

Here the last equality is since we set $k = 4T + 2$. Also, recalling that $\mathcal{M}$ starts at $\boldsymbol{x}_0 = 0$ we have

$$\|\boldsymbol{x}_0 - \boldsymbol{z}^*\|^2 = \|\boldsymbol{x}^*\|^2 = c^2 \sum_{i=1}^{4T+2} i^2 = \frac{c^2}{6}(4T + 2)(4T + 3)(8T + 5).$$

Summarizing the above two relations we have

$$\|\boldsymbol{x}_T - \boldsymbol{z}^*\|^2 > \frac{1}{8}\|\boldsymbol{x}_0 - \boldsymbol{z}^*\|^2. \tag{3.11}$$

Furthermore, applying (3.11), Lemma 2.4, and the estimate of lower bound in (2.18) to (3.10), we have

$$\begin{aligned}
&l_{A_k U, \boldsymbol{b}_k}(\boldsymbol{x}_T) - l^*_{A_k U, \boldsymbol{b}_k} \\
&\geq 4(k - 2T - 1)\left[(\sigma - \zeta)c - \log\cosh(\sigma c) - \log\cosh(\zeta c)\right] \\
&\geq 2(k - 2T - 1)c^2\sigma^2 \\
&= \frac{6 * (k - 2T - 1)\sigma^2\|\boldsymbol{x}_0 - \boldsymbol{z}^*\|^2}{(2T + 1)(4T + 3)(8T + 5)}.
\end{aligned}$$

Applying the estimate of $\|A_k\|$ in (2.4) to the above, and recalling that $k = 4T + 2$, we obtain

$$l_{A_k U, \boldsymbol{b}_k}(\boldsymbol{x}_T) - l^*_{A_k U, \boldsymbol{b}_k} \leq \frac{3\|A\|^2\|\boldsymbol{x}_0 - \boldsymbol{z}^*\|^2}{32(4T + 3)(8T + 5)}.$$

By setting $A := A_k U \in \mathbb{R}^{(16T+8)\times(4T+2)}$ and $b := \boldsymbol{b}_k \in \mathbb{R}^{(16T+8)}$, we conclude the proof from (3.10) and (3.11). ∎

4. Concluding Remarks

In this paper, we describe some worst-case datasets for deterministic first-order methods on solving binary logistic regression. The binary logistic regression functions with our worst-case datasets can also server as new worst-case function instances among the class of smooth convex optimization problems.

It should be noted that our description of A_k and $\boldsymbol{b}_k$ in (2.2) are designed so that the optimal intercept of binary logistic regression is 0. If we are focusing only on homogeneous linear predictor case without requiring the optimal intercept to be 0, an easier dataset can be designed by simply setting

$$A_k := \begin{pmatrix} 2\sigma W_k \\ 2\zeta W_k \end{pmatrix} \in \mathbb{R}^{2k\times k}, \quad \boldsymbol{b}_k := \begin{pmatrix} \boldsymbol{1}_k \\ -\boldsymbol{1}_k \end{pmatrix} \in \mathbb{R}^{2k}$$

and follow the derivations in Sections 2 and 3.

References

Francis Bach. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4:384–414, 2010.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *arXiv preprint arXiv:1710.11606*, 2017a.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points II: First-order methods. *arXiv preprint arXiv:1711.00841*, 2017b.

Jelena Diakonikolas and Cristóbal Guzmán. Lower bounds for parallel and randomized convex optimization. *arXiv preprint arXiv:1811.01903*, 2018.

Yoel Drori. The exact information-based complexity of smooth convex minimization. *Journal of Complexity*, 39:1–16, 2017.

Cristóbal Guzmán and Arkadi Nemirovski. On lower complexity bounds for large-scale smooth convex optimization. *Journal of Complexity*, 31(1):1–14, 2015.

Anatoli Juditsky and Yuri Nesterov. Deterministic and stochastic primal-dual subgradient algorithms for uniformly convex minimization. *Stochastic Systems*, 4(1):44–80, 2014.

A. Nemirovski and D. Yudin. *Problem complexity and method efficiency in optimization.* Wiley-Interscience Series in Discrete Mathematics. John Wiley, XV, 1983.

A. S. Nemirovski. Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2):153–175, 1992.

Y. E. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* Kluwer Academic Publishers, Massachusetts, 2004.

Yuyuan Ouyang and Yangyang Xu. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *Mathematical Programming*, pages 1–35, 2019.

Blake E Woodworth and Nati Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in neural information processing systems*, pages 3639–3647, 2016.