Accelerated Stochastic Gradient-free and Projection-free Methods

Feihu Huang [1 2] **Lue Tao** [1 2] **Songcan Chen** [1 2]

Abstract

In the paper, we propose a class of accelerated stochastic gradient-free and projection-free (a.k.a., zeroth-order Frank-Wolfe) methods to solve the constrained stochastic and finite-sum nonconvex optimization. Specifically, we propose an accelerated stochastic zeroth-order Frank-Wolfe (Acc-SZOFW) method based on the variance reduced technique of SPIDER/SpiderBoost and a novel momentum accelerated technique. Moreover, under some mild conditions, we prove that the Acc-SZOFW has the function query complexity of $O(d\sqrt{n}\epsilon^{-2})$ for finding an ϵ-stationary point in the finite-sum problem, which improves the exiting best result by a factor of $O(\sqrt{n}\epsilon^{-2})$, and has the function query complexity of $O(d\epsilon^{-3})$ in the stochastic problem, which improves the exiting best result by a factor of $O(\epsilon^{-1})$. To relax the large batches required in the Acc-SZOFW, we further propose a novel accelerated stochastic zeroth-order Frank-Wolfe (Acc-SZOFW*) based on a new variance reduced technique of STORM, which still reaches the function query complexity of $O(d\epsilon^{-3})$ in the stochastic problem without relying on any large batches. In particular, we present an accelerated framework of the Frank-Wolfe methods based on the proposed momentum accelerated technique. The extensive experimental results on black-box adversarial attack and robust black-box classification demonstrate the efficiency of our algorithms.

[1]College of Computer Science & Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China [2]MIIT Key Laboratory of Pattern Analysis & Machine Intelligence. Correspondence to: Feihu Huang <huangfeihu@nuaa.edu.cn>.

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

1. Introduction

In the paper, we focus on solving the following constrained stochastic and finite-sum optimization problems

$$\min_{x\in\mathcal{X}} f(x) = \begin{cases} \mathbb{E}_{\xi}[f(x;\xi)] & \text{(stochastic)} \\ \frac{1}{n}\sum_{i=1}^{n} f_i(x) & \text{(finite-sum)} \end{cases} \tag{1}$$

where $f(x) : \mathbb{R}^d \to \mathbb{R}$ is a nonconvex and smooth loss function, and the restricted domain $\mathcal{X} \subseteq \mathbb{R}^d$ is supposed to be convex and compact, and ξ is a random variable that following an unknown distribution. When $f(x)$ denotes the expected risk function, the problem (1) will be seen as a stochastic problem. While $f(x)$ denotes the empirical risk function, it will be seen as a finite-sum problem. In fact, the problem (1) appears in many machine learning models such as multitask learning, recommendation systems and, structured prediction (Jaggi, 2013; Lacoste-Julien et al., 2013; Hazan & Luo, 2016). For solving the constrained problem (1), one common approach is the projected gradient method (Iusem, 2003) that alternates between optimizing in the unconstrained space and projecting onto the constrained set $\mathcal{X}$. However, the projection is quite expensive to compute in many constrained sets such as the set of all bounded nuclear norm matrices. The Frank-Wolfe algorithm (i.e., conditional gradient)(Frank & Wolfe, 1956; Jaggi, 2013) is a good candidate for solving the problem (1), which only needs to compute a linear operator instead of projection operator at each iteration. Following (Jaggi, 2013), the linear optimization on $\mathcal{X}$ is much faster than the projection onto $\mathcal{X}$ in many problems such as the set of all bounded nuclear norm matrices.

Due to its projection-free property and ability to handle structured constraints, the Frank-Wolfe algorithm has recently regained popularity in many machine learning applications, and its variants have been widely studied. For example, several convex variants of Frank-Wolfe algorithm (Jaggi, 2013; Lacoste-Julien & Jaggi, 2015; Lan & Zhou, 2016; Xu & Yang, 2018) have been studied. In the big data setting, the corresponding online and stochastic Frank-Wolfe algorithms (Hazan & Kale, 2012; Hazan & Luo, 2016; Hassani et al., 2019; Xie et al., 2019) have been developed, and their convergence rates were studied. The above Frank-Wolfe algorithms were mainly studied in the convex setting.

Table 1. Function query complexity comparison of the representative non-convex zeroth-order *Frank-Wolfe* methods for finding an ϵ-stationary point of the problem (1), *i.e.,* $\mathbb{E}\|\nabla \mathcal{G}(x)\| \leq \epsilon$. T denotes the total iterations. GauGE, UniGE and CooGE are abbreviations of Gaussian distribution, Uniform distribution and Coordinate-wise smoothing gradient estimators, respectively. Note that FW-Black and Acc-ZO-FW are deterministic algorithms, the other are stochastic algorithms. Here **query-size** denotes the *function query size* required in estimating one zeroth-order gradient in these algorithms. Note that these query-sizes are only used in the theoretical analysis.

Problem	Algorithm	Reference	Gradient Estimator	Query Complexity	Query-Size
Finite-Sum	FW-Black	Chen et al. (2018)	GauGE or UniGE	$O(dn\epsilon^{-4})$	$O(ndT)$
	Acc-ZO-FW	Ours	CooGE	$O(dn\epsilon^{-2})$	$O(nd)$
	Acc-SZOFW	Ours	CooGE	$O(dn^{\frac{1}{2}}\epsilon^{-2})$	$O(n^{\frac{1}{2}}d)$
Stochastic	ZO-SFW	Sahu et al. (2019)	GauGE	$O(d^{\frac{4}{3}}\epsilon^{-4})$	$O(1)$
	ZSCG	Balasubramanian & Ghadimi (2018)	GauGE	$O(d\epsilon^{-4})$	$O(dT)$
	Acc-SZOFW	Ours	CooGE	$O(d\epsilon^{-3})$	$O(dT^{1/2})$
	Acc-SZOFW	Ours	UniGE	$O(d\epsilon^{-3})$	$O(d^{-1/2}T^{1/2})$
	Acc-SZOFW*	Ours	CooGE	$O(d\epsilon^{-3})$	$O(d)$
	Acc-SZOFW*	Ours	UniGE	$O(d^{\frac{4}{3}}\epsilon^{-3})$	$O(1)$

In fact, the Frank-Wolfe algorithm and its variants are also successful in solving nonconvex problems such as adversarial attacks (Chen et al., 2018). Recently, some nonconvex variants of Frank-Wolfe algorithm (Lacoste-Julien, 2016; Reddi et al., 2016; Qu et al., 2018; Shen et al., 2019; Yurtsever et al., 2019; Hassani et al., 2019; Zhang et al., 2019) have been developed.

Until now, the above Frank-Wolfe algorithm and its variants need to compute the gradients of objective functions at each iteration. However, in many complex machine learning problems, the explicit gradients of the objective functions are difficult or infeasible to obtain. For example, in the reinforcement learning (Malik et al., 2019; Huang et al., 2020), some complex graphical model inference (Wainwright et al., 2008) and metric learning (Chen et al., 2019a) problems, it is difficult to compute the explicit gradients of objective functions. Even worse, in the black-box adversarial attack problems (Liu et al., 2018b; Chen et al., 2018), only function values (*i.e.*, prediction labels) are accessible. Clearly, the above Frank-Wolfe methods will fail in dealing with these problems. Since it only uses the function values in optimization, the gradient-free (zeroth-order) optimization method (Duchi et al., 2015; Nesterov & Spokoiny, 2017) is a promising choice to address these problems. More recently, some zeroth-order Frank-Wolfe methods (Balasubramanian & Ghadimi, 2018; Chen et al., 2018; Sahu et al., 2019) have been proposed and studied. However, these zeroth-order Frank-Wolfe methods suffer from high function query complexity in solving the problem (1) (please see Table 1).

In the paper, thus, we propose a class of accelerated zeroth-order Frank-Wolfe methods to solve the problem (1), where $f(x)$ is possibly black-box. Specifically, we propose an accelerated stochastic zeroth-order Frank-Wolfe (Acc-SZOFW) method based on the variance reduced technique of SPIDER/SpiderBoost (Fang et al., 2018; Wang et al., 2018) and a novel momentum accelerated technique. Further, we propose a novel accelerated stochastic zeroth-order Frank-Wolfe (Acc-SZOFW*) to relax the large mini-batch size required in the Acc-SZOFW.

Contributions

In summary, our main contributions are given as follows:

1) We propose an accelerated stochastic zeroth-order Frank-Wolfe (Acc-SZOFW) method based on the variance reduced technique of SPIDER/SpiderBoost and a novel momentum accelerated technique.
2) Moreover, under some mild conditions, we prove that the Acc-SZOFW has the function query complexity of $O(d\sqrt{n}\epsilon^{-2})$ for finding an ϵ-stationary point in the finite-sum problem (1), which improves the exiting best result by a factor of $O(\sqrt{n}\epsilon^{-2})$, and has the function query complexity of $O(d\epsilon^{-3})$ in the stochastic problem (1), which improves the exiting best result by a factor of $O(\epsilon^{-1})$.
3) We further propose a novel accelerated stochastic zeroth-order Frank-Wolfe (Acc-SZOFW*) to relax the large mini-batch size required in the Acc-SZOFW. We prove that the Acc-SZOFW* still has the function query complexity of $O(d\epsilon^{-3})$ without relying on the large batches.
4) In particular, we propose an accelerated framework of the Frank-Wolfe methods based on the proposed momentum accelerated technique.

2. Related Works

2.1. Zeroth-Order Methods

Zeroth-order (gradient-free) methods can be effectively used to solve many machine learning problems, where the explicit gradient is difficult or infeasible to obtain. Recently, the zeroth-order methods have been widely studied in machine learning community. For example, several zeroth-order methods (Ghadimi & Lan, 2013; Duchi et al., 2015;

Nesterov & Spokoiny, 2017) have been proposed by using the Gaussian smoothing technique. Subsequently, (Liu et al., 2018b; Ji et al., 2019) recently proposed accelerated zeroth-order stochastic gradient methods based on the variance reduced techniques. To deal with nonsmooth optimization, some zeroth-order proximal gradient methods (Ghadimi et al., 2016; Huang et al., 2019c; Ji et al., 2019) and zeroth-order ADMM-based methods (Gao et al., 2018; Liu et al., 2018a; Huang et al., 2019a;b) have been proposed. In addition, more recently, (Chen et al., 2019b) has proposed a zeroth-order adaptive momentum method. To solve the constrained optimization, the zeroth-order Frank-Wolfe methods (Balasubramanian & Ghadimi, 2018; Chen et al., 2018; Sahu et al., 2019) and the zeroth-order projected gradient methods (Liu et al., 2018c) have been recently proposed and studied.

2.2. Variance-Reduced and Momentum Methods

To accelerate stochastic gradient descent (SGD) algorithm, various variance-reduced algorithms such as SAG (Roux et al., 2012), SAGA (Defazio et al., 2014), SVRG (Johnson & Zhang, 2013) and SARAH (Nguyen et al., 2017a) have been presented and studied. Due to the popularity of deep learning, recently the large-scale nonconvex learning problems received wide interest in machine learning community. Thus, recently many corresponding variance-reduced algorithms to nonconvex SGD have also been proposed and studied, e.g., SVRG (Allen-Zhu & Hazan, 2016; Reddi et al., 2016), SCSG (Lei et al., 2017), SARAH (Nguyen et al., 2017b), SPIDER (Fang et al., 2018), SpiderBoost (Wang et al., 2018; 2019), SNVRG (Zhou et al., 2018).

Another effective alternative is to use momentum-based method to accelerate SGD. Recently, various momentum-based stochastic algorithms for the convex optimization have been proposed and studied, e.g., APCG (Lin et al., 2014), AccProxSVRG (Nitanda, 2014) and Katyusha (Allen-Zhu, 2017). At the same time, for the nonconvex optimization, some momentum-based stochastic algorithms have been also studied, e.g., RSAG (Ghadimi & Lan, 2016), ProxSpiderBoost-M (Wang et al., 2019), STORM (Cutkosky & Orabona, 2019) and Hybrid-SGD (Tran-Dinh et al., 2019).

3. Preliminaries

3.1. Zeroth-Order Gradient Estimators

In this subsection, we introduce two useful zeroth-order gradient estimators, i.e., uniform smoothing gradient estimator (UniGE) and coordinate smoothing gradient estimator (CooGE) (Liu et al., 2018b; Ji et al., 2019). Given any function $f_i(x) : \mathbb{R}^d \to \mathbb{R}$, the UniGE can generate an approximated gradient as follows:

$$\hat{\nabla}_{uni} f_i(x) = \frac{d(f_i(x+\beta u) - f_i(x))}{\beta} u, \tag{2}$$

where $u \in \mathbb{R}^d$ is a vector generated from the uniform distribution over the unit sphere, and β is a smoothing parameter. While the CooGE can generate an approximated gradient:

$$\hat{\nabla}_{coo} f_i(x) = \sum_{j=1}^{d} \frac{f_i(x+\mu_j e_j) - f_i(x-\mu_j e_j)}{2\mu_j} e_j, \tag{3}$$

where μ_j is a coordinate-wise smoothing parameter, and e_j is a basis vector with 1 at its j-th coordinate, and 0 otherwise. Without loss of generality, let $\mu = \mu_1 = \cdots = \mu_d$.

3.2. Standard Frank-Wolfe Algorithm and Assumptions

The standard Frank-Wolfe (i.e., conditional gradient) algorithm solves the above problem (1) by the following iteration: at $t+1$-th iteration,

$$\begin{cases} w_{t+1} = \arg\max_{w \in \mathcal{X}} \langle w, -\nabla f(x_t) \rangle, \\ x_{t+1} = (1-\gamma_{t+1}) x_t + \gamma_{t+1} w_{t+1}, \end{cases} \tag{4}$$

where $\gamma_{t+1} \in (0,1)$ is a step size. For the nonconvex optimization, we apply the following duality gap (i.e., Frank-Wolfe gap (Jaggi, 2013))

$$\mathcal{G}(x) = \max_{w \in \mathcal{X}} \langle w - x, -\nabla f(x) \rangle, \tag{5}$$

to give the standard criteria of convergence $\|\mathcal{G}(x)\| \leq \epsilon$ (or $\mathbb{E}\|\mathcal{G}(x)\| \leq \epsilon$) for finding an ϵ-stationary point, as in (Reddi et al., 2016).

Next, we give some standard assumptions regarding problem (1) as follows:

Assumption 1. *Let $f_i(x) = f(x;\xi_i)$, where ξ_i samples from the distribution of random variable ξ. Each loss function $f_i(x)$ is L-smooth such that*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x-y\|, \ \forall x, y \in \mathcal{X},$$
$$f_i(x) \leq f_i(y) + \nabla f_i(y)^T (x-y) + \frac{L}{2}\|x-y\|^2.$$

Let $f_\beta(x) = \mathbb{E}_{u \sim U_B}[f(x+\beta u)]$ be a smooth approximation of $f(x)$, where U_B is the uniform distribution over the d-dimensional unit Euclidean ball B. Following (Ji et al., 2019), we have $\mathbb{E}_{(u,\xi)}[\hat{\nabla}_{uni} f_\xi(x)] = \nabla f_\beta(x)$.

Assumption 2. *The variance of stochastic (zeroth-order) gradient is bounded,* i.e., *there exists a constant $\sigma_1 > 0$ such that for all x, it follows $\mathbb{E}\|\nabla f_\xi(x) - \nabla f(x)\|^2 \leq \sigma_1^2$; There exists a constant $\sigma_2 > 0$ such that for all x, it follows $\mathbb{E}\|\hat{\nabla}_{uni} f_\xi(x) - \nabla f_\beta(x)\|^2 \leq \sigma_2^2$.*

Assumption 3. *The constraint set $\mathcal{X} \subseteq \mathbb{R}^d$ is compact with the diameter:* $\max_{x,y\in\mathcal{X}} \|x-y\| \leq D$.

Assumption 4. *The objective function $f(x)$ is bounded from below in $\mathcal{X}$,* i.e., *there exists a non-negative constant $\triangle$, for all $x \in \mathcal{X}$ such as $f(x) - \inf_{y\in\mathcal{X}} f(y) \leq \triangle$.*

Assumption 1 imposes the smoothness on each loss function $f_i(x)$ or $f(x,\xi_i)$, which is commonly used in the convergence analysis of nonconvex algorithms (Ghadimi et al., 2016). Assumption 2 shows that the variance of stochastic or zeroth-order gradient is bounded in norm, which have been commonly used in the convergence analysis of stochastic zeroth-order algorithms (Gao et al., 2018; Ji et al., 2019). Assumptions 3 and 4 are standard for the convergence analysis of Frank-Wolfe algorithms (Jaggi, 2013; Shen et al., 2019; Yurtsever et al., 2019).

4. Accelerated Stochastic Zeroth-Order Frank-Wolfe Algorithms

In the section, we first propose an accelerated stochastic zeroth-order Frank-Wolfe (Acc-SZOFW) algorithm based on the variance reduced technique of SPIDER/SpiderBoost and a novel momentum accelerated technique. We then further propose a novel accelerated stochastic zeroth-order Frank-Wolfe (Acc-SZOFW*) algorithm to relax the large mini-batch size required in the Acc-SZOFW.

4.1. Acc-SZOFW Algorithm

In the subsection, we propose an Acc-SZOFW algorithm to solve the problem (1), where the loss function is possibly black-box. The Acc-SZOFW algorithm is given in Algorithm 1.

We first propose an accelerated **deterministic** zeroth-order Frank-Wolfe (Acc-ZO-FW) algorithm to solve the finite-sum problem (1) as a baseline by using the zeroth-order gradient $v_t = \frac{1}{n}\sum_{i=1}^n \hat{\nabla}_{coo} f_i(z_t)$ in Algorithm 1. Although Chen et al. (2018) has proposed a **deterministic** zeroth-order Frank-Wolfe (FW-Black) algorithm by using the momentum-based accelerated zeroth-order gradients, our Acc-ZO-FW algorithm still has lower query complexity than the FW-Black algorithm (see Table 1).

When the sample size n is very large in the finite-sum optimization problem (1), we will need to waste lots of time to obtain the estimated full gradient of $f(x)$, and in turn make the whole algorithm became very slow. Even worse, for the stochastic optimization problem (1), we can never obtain the estimated full gradient of $f(x)$. As a result, the stochastic optimization method is a good choice. Specifically, we can draw a mini-batch $\mathcal{B} \subseteq \{1,2,\cdots,n\}$ $(b = |\mathcal{B}|)$ or $\mathcal{B} = \{\xi_1,\cdots,\xi_b\}$ from the distribution of random variable ξ, and can obtain the following stochastic zeroth-order gradient:

$$\hat{\nabla} f_{\mathcal{B}}(x) = \frac{1}{b}\sum_{j\in\mathcal{B}} \hat{\nabla} f_j(x),$$

where $\hat{\nabla} f_j(\cdot)$ includes $\hat{\nabla}_{coo} f_j(\cdot)$ and $\hat{\nabla}_{uni} f_j(\cdot)$.

However, this standard zeroth-order stochastic Frank-Wolfe algorithm suffers from large variance in the zeroth-order stochastic gradient. Following (Balasubramanian & Ghadimi, 2018; Sahu et al., 2019), this variance will result in high function query complexity. Thus, we use the variance reduced technique of SPIDER/SpiderBoost as in (Ji et al., 2019) to reduce the variance in the stochastic gradients. Specifically, in Algorithm 1, we use the following semi-stochastic gradient for solving the stochastic problem:

$$v_t = \begin{cases} \dfrac{1}{b_1}\displaystyle\sum_{i\in\mathcal{B}_1} \hat{\nabla} f_i(x_t), & \text{if} \quad \text{mod}\ (t,q) = 0 \\ \dfrac{1}{b_2}\displaystyle\sum_{i\in\mathcal{B}_2} \big(\hat{\nabla} f_i(x_t) - \hat{\nabla} f_i(x_{t-1})\big) + v_{t-1}, & \text{otherwise} \end{cases}$$

Algorithm 1 Acc-SZOFW Algorithm

Input: Total iteration T, step-sizes $\{\eta_t, \gamma_t \in (0,1)\}_{t=0}^{T-1}$, weighted parameters $\{\alpha_t \in [0,1]\}_{t=0}^{T-1}$, epoch-size q, mini-batch size b or b_1, b_2;
Initialize: $x_0 = y_0 = z_0 \in \mathcal{X}$;
for $t = 0, 1, \ldots, T-1$ **do**
  if $\text{mod}\ (t,q) = 0$ **then**
    For the **finite-sum** setting, compute $v_t = \hat{\nabla}_{coo} f(z_t) = \frac{1}{n}\sum_{i=1}^n \hat{\nabla}_{coo} f_i(z_t)$;
    For the **stochastic** setting, randomly select b_1 samples $\mathcal{B}_1 = \{\xi_1,\cdots,\xi_{b_1}\}$, and compute $v_t = \hat{\nabla}_{coo} f_{\mathcal{B}_1}(z_t)$, or draw i.i.d. $\{u_1,\cdots,u_{b_1}\}$ from uniform distribution over unit sphere, then compute $v_t = \hat{\nabla}_{uni} f_{\mathcal{B}_1}(z_t)$;
  else
    For the **finite-sum** setting, randomly select $b = |\mathcal{B}|$ samples $\mathcal{B} \subseteq \{1,\cdots,n\}$, and compute $v_t = \frac{1}{b}\sum_{j\in\mathcal{B}}[\hat{\nabla}_{coo} f_j(z_t) - \hat{\nabla}_{coo} f_j(z_{t-1})] + v_{t-1}$;
    For the **stochastic** setting, randomly select b_2 samples $\mathcal{B}_2 = \{\xi_1,\cdots,\xi_{b_2}\}$, and compute $v_t = \frac{1}{b_2}\sum_{j\in\mathcal{B}_2}[\hat{\nabla}_{coo} f_j(z_t) - \hat{\nabla}_{coo} f_j(z_{t-1})] + v_{t-1}$, or draw i.i.d. $\{u_1,\cdots,u_{b_2}\}$ from uniform distribution over unit sphere, then $v_t = \frac{1}{b_2}\sum_{j\in\mathcal{B}_2}[\hat{\nabla}_{uni} f_j(z_t) - \hat{\nabla}_{uni} f_j(z_{t-1})] + v_{t-1}$;
  end if
  Optimize $w_t = \arg\max_{w\in\mathcal{X}} \langle w, -v_t\rangle$;
  Update $x_{t+1} = x_t + \gamma_t(w_t - x_t)$;
  Update $y_{t+1} = z_t + \eta_t(w_t - z_t)$;
  Update $z_{t+1} = (1-\alpha_{t+1})y_{t+1} + \alpha_{t+1}x_{t+1}$;
end for
Output: z_ζ chosen uniformly random from $\{z_t\}_{t=1}^T$.

Algorithm 2 Acc-SZOFW* Algorithm

Input: Total iteration T, step-sizes $\{\eta_t, \gamma_t \in (0,1)\}_{t=0}^{T-1}$, weighted parameters $\{\alpha_t \in [0,1]\}_{t=1}^{T-1}$ and the parameter $\{\rho_t\}_{t=1}^{T-1}$;
Initialize: $x_0 = y_0 = z_0 \in \mathcal{X}$;
for $t = 0, 1, \ldots, T-1$ **do**
  if $t = 0$ **then**
    Sample a point ξ_0, and compute $v_0 = \hat{\nabla}_{coo} f_{\xi_0}(z_0)$, or draw a vector $u \in \mathbb{R}^d$ from uniform distribution over unit sphere, then compute $v_0 = \hat{\nabla}_{uni} f_{\xi_0}(z_0)$;
  else
    Sample a point ξ_t, and compute $v_t = \hat{\nabla}_{coo} f_{\xi_t}(z_t) + (1-\rho_t)\big(v_{t-1} - \hat{\nabla}_{coo} f_{\xi_t}(z_{t-1})\big)$, or draw a vector $u \in \mathbb{R}^d$ from uniform distribution over unit sphere, then compute $v_t = \hat{\nabla}_{uni} f_{\xi_t}(z_t) + (1-\rho_t)\big(v_{t-1} - \hat{\nabla}_{uni} f_{\xi_t}(z_{t-1})\big)$;
  end if
  Optimize $w_t = \arg\max_{w \in \mathcal{X}} \langle w, -v_t \rangle$;
  Update $x_{t+1} = x_t + \gamma_t(w_t - x_t)$;
  Update $y_{t+1} = z_t + \eta_t(w_t - z_t)$;
  Update $z_{t+1} = (1-\alpha_{t+1})y_{t+1} + \alpha_{t+1}x_{t+1}$;
end for
Output: z_ζ chosen uniformly random from $\{z_t\}_{t=1}^{T}$.

Moreover, we propose a novel momentum accelerated framework for the Frank-Wolfe algorithm. Specifically, we introduce two intermediate variables x and y, as in (Wang et al., 2019), and our algorithm keeps all variables $\{x, y, z\}$ in the constraint set $\mathcal{X}$. In Algorithm 1, when set $\alpha_{t+1} = 0$ or $\alpha_{t+1} = 1$, our algorithm will reduce to the zeroth-order Frank-Wolfe algorithm with the variance reduced technique of SPIDER/SpiderBoost. When $\alpha_{t+1} \in (0,1)$, our algorithm will generate the following iterations:

$$
\begin{aligned}
z_1 &= z_0 + ((1-\alpha_1)\eta_0 + \alpha_1\gamma_0)(w_0 - z_0),\\
z_2 &= z_1 + ((1-\alpha_2)\eta_1 + \alpha_2\gamma_1)(w_1 - z_1)\\
&\quad + \alpha_2(1-\gamma_1)(1-\alpha_1)(\gamma_0 - \eta_0)(w_0 - z_0),\\
z_3 &= z_2 + ((1-\alpha_3)\eta_2 + \alpha_3\gamma_2)(w_2 - z_2)\\
&\quad + \alpha_3(1-\gamma_2)(1-\alpha_2)(\gamma_1 - \eta_1)(w_1 - z_1)\\
&\quad + \alpha_3(1-\gamma_2)(1-\alpha_2)(1-\gamma_1)(1-\alpha_1)(\gamma_0 - \eta_0)(w_0 - z_0),\\
&\quad \ldots
\end{aligned}
$$

From the above iterations, the updating parameter z_t is a linear combination of the previous terms $w_i - z_i$ $(i \leq t)$, which coincides the aim of momentum accelerated technique (Nesterov, 2004; Allen-Zhu, 2017). In fact, our momentum accelerated technique does not rely on the version of gradient v_t. In other words, our momentum accelerated technique can be applied in the zeroth-order, first-order, determinate or stochastic Frank-Wolfe algorithms.

4.2. Acc-SZOFW* Algorithm

In this subsection, we propose a novel Acc-SZOFW* algorithm based on a new momentum-based variance reduced technique of STORM/Hybrid-SGD (Cutkosky & Orabona, 2019; Tran-Dinh et al., 2019). Although the above Acc-SZOFW algorithm reaches a lower function query complexity, it requires large batches (Please see Table 1). Clearly, the Acc-SZOFW algorithm can not be well competent to the very large-scale problems and the data flow problems. Thus, we further propose a novel Acc-SZOFW* algorithm to relax the large batches required in the Acc-SZOFW. Algorithm 2 details the Acc-SZOFW* algorithm.

In Algorithm 2, we apply the variance-reduced technique of STORM to estimate the zeroth-order stochastic gradients, and update the parameters $\{x, y, z\}$ as in Algorithm 1. Specifically, we use the zeroth-order stochastic gradients as follows:

$$
\begin{aligned}
v_t &= \rho_t \underbrace{\hat{\nabla} f_{\xi_t}(z_t)}_{\text{SGD}} + (1-\rho_t)\big(\underbrace{\hat{\nabla} f_{\xi_t}(z_t) - \hat{\nabla} f_{\xi_t}(z_{t-1}) + v_{t-1}}_{\text{SPIDER}}\big)\\
&= \hat{\nabla} f_{\xi_t}(z_t) + (1-\rho_t)\big(v_{t-1} - \hat{\nabla} f_{\xi_t}(z_{t-1})\big), \qquad (6)
\end{aligned}
$$

where $\rho_t \in (0,1]$. Recently, (Zhang et al., 2019; Xie et al., 2019) have been applied this variance-reduced technique of STORM to the Frank-Wolfe algorithms. However, these algorithms strictly rely on the unbiased stochastic gradient. To the best of our knowledge, we are the first to apply the STORM to the zeroth-order algorithm, which does not rely on the unbiased stochastic gradient.

5. Convergence Analysis

In the section, we study the convergence properties of both the Acc-SZOFW and Acc-SZOFW* algorithms. *All related proofs are provided in the supplementary document.* Throughout the paper, $\|\cdot\|$ denotes the vector ℓ_2 norm and the matrix spectral norm, respectively. Without loss of generality, let $\alpha_t = \frac{1}{t+1}$, $\gamma_t = (1+\theta_t)\eta_t$ with $\theta_t = \frac{1}{(t+1)(t+2)}$ in our algorithms.

5.1. Convergence Properties of Acc-SZOFW Algorithm

In this subsection, we study the convergence properties of the Acc-SZOFW Algorithm based on the CooGE and UniGE zeroth-order gradients, respectively. The detailed proofs are provided in the Appendix A.1.

We first study the convergence properties of the deterministic **Acc-ZO-FW** algorithm as a baseline, which is Algorithm 1 using the **deterministic** zeroth-order gradient $v_t = \frac{1}{n}\sum_{i=1}^{n} \hat{\nabla}_{coo} f_i(z_t)$ for solving the **finite-sum** problem (1).

5.1.1. DETERMINISTIC ACC-ZO-FW ALGORITHM

Theorem 1. *Suppose $\{x_t, y_t, z_t\}_{t=0}^{T-1}$ be generated from Algorithm 1 by using the **deterministic** zeroth-order gradient $v_t = \frac{1}{n}\sum_{i=1}^{n} \hat{\nabla}_{coo} f_i(z_t)$, and let $\alpha_t = \frac{1}{t+1}$, $\theta_t = \frac{1}{(t+1)(t+2)}$, $\gamma_t = (1+\theta_t)\eta_t$, $\eta = \eta_t = T^{-\frac{1}{2}}$, $\mu = d^{-\frac{1}{2}}T^{-\frac{1}{2}}$, then we have*

$$\mathbb{E}[\mathcal{G}(z_\zeta)] = \frac{1}{T}\sum_{t=1}^{T-1} \mathcal{G}(z_t) \leq O(\frac{1}{T^{\frac{1}{2}}}) + O(\frac{\ln(T)}{T^{\frac{3}{2}}}),$$

where z_ζ is chosen uniformly randomly from $\{z_t\}_{t=0}^{T-1}$.

Remark 1. *Theorem 1 shows that the deterministic Acc-ZO-FW algorithm under the **CooGE** has $O(T^{-\frac{1}{2}})$ convergence rate. The Acc-ZO-FW algorithm needs nd samples to estimate the zeroth-order gradient v_t at each iteration. For finding an ϵ-stationary point, i.e., $\mathbb{E}[\mathcal{G}(z_\zeta)] \leq \epsilon$, by $T^{-\frac{1}{2}} \leq \epsilon$, we choose $T = \epsilon^{-2}$. Thus the **deterministic** Acc-ZO-FW has the function query complexity of $ndT = O(dn\epsilon^{-2})$. Comparing with the existing deterministic zeroth-order Frank-Wolfe algorithm,* i.e., *FW-Black (Chen et al., 2018), our Acc-ZO-FW algorithm has a lower query complexity of $ndT = O(dn\epsilon^{-2})$, which improves the existing result by a factor of $O(\epsilon^{-2})$ (please see Table 1).*

5.1.2. ACC-SZOFW (COOGE) ALGORITHM

Lemma 1. *Suppose the zeroth-order stochastic gradient v_t be generated from Algorithm 1 by using the **CooGE** zeroth-order gradient estimator. Let $\alpha_t = \frac{1}{t+1}$, $\theta_t = \frac{1}{(t+1)(t+2)}$ and $\gamma_t = (1+\theta_t)\eta_t$ in Algorithm 1. For the **finite-sum** setting, we have*

$$\mathbb{E}\|\nabla f(z_t) - v_t\| \leq L\sqrt{d}\mu + \frac{L(\sqrt{6d}\mu + 2\sqrt{3D}\eta)}{\sqrt{b/q}}.$$

*For the **stochastic** setting, we have*

$$\mathbb{E}\|\nabla f(z_t) - v_t\| \leq L\sqrt{d}\mu + \frac{L(\sqrt{6d}\mu + 2\sqrt{3D}\eta)}{\sqrt{b_2/q}} + \frac{\sqrt{3}\sigma_1}{\sqrt{b_1}} + \sqrt{6d}L\mu.$$

Theorem 2. *Suppose $\{x_t, y_t, z_t\}_{t=0}^{T-1}$ be generated from Algorithm 1 by using the **CooGE** zeroth-order gradient estimator, and let $\alpha_t = \frac{1}{t+1}$, $\theta_t = \frac{1}{(t+1)(t+2)}$, $\gamma_t = (1+\theta_t)\eta_t$, $\eta = \eta_t = T^{-\frac{1}{2}}$, $\mu = d^{-\frac{1}{2}}T^{-\frac{1}{2}}$, $b = q$, or $b_2 = q$ and $b_1 = T$, then we have*

$$\mathbb{E}[\mathcal{G}(z_\zeta)] = \frac{1}{T}\sum_{t=1}^{T-1} \mathbb{E}[\mathcal{G}(z_t)] \leq O(\frac{1}{T^{\frac{1}{2}}}) + O(\frac{\ln(T)}{T^{\frac{3}{2}}}),$$

where z_ζ is chosen uniformly randomly from $\{z_t\}_{t=0}^{T-1}$.

Remark 2. *Theorem 2 shows that the Acc-SZOFW (CooGE) algorithm has convergence rate of $O(T^{-\frac{1}{2}})$. When $\mod(t,q) = 0$, the Acc-SZOFW algorithm needs nd or $b_1 d$ samples to estimate the zeroth-order gradient v_t at each iteration and needs T/q iterations, otherwise it needs $2bd$ or $2b_2 d$ samples to estimate v_t at each iteration and needs T iterations. In the **finite-sum** setting, by $T^{-\frac{1}{2}} \leq \epsilon$, we choose $T = \epsilon^{-2}$, and let $b = q = \sqrt{n}$, the Acc-SZOFW has the function query complexity of $dnT/q + 2dbT = O(d\sqrt{n}\epsilon^{-2})$ for finding an ϵ-stationary point. In the **stochastic** setting, let $b_2 = q = \epsilon^{-1}$ and $b_1 = T = \epsilon^{-2}$, the Acc-SZOFW has the function query complexity of $db_1 T/q + 2db_2 T = O(d\epsilon^{-3})$ for finding an ϵ-stationary point.*

5.1.3. ACC-SZOFW (UNIGE) ALGORITHM

Lemma 2. *Suppose the zeroth-order stochastic gradient v_t be generated from Algorithm 1 by using the **UniGE** zeroth-order gradient estimator. Let $\alpha_t = \frac{1}{t+1}$, $\theta_t = \frac{1}{(t+1)(t+2)}$ and $\gamma_t = (1+\theta_t)\eta_t$ in Algorithm 1. For the **stochastic** setting, we have*

$$\mathbb{E}\|\nabla f(z_t) - v_t\| \leq \frac{\beta L d}{2} + \frac{L(\sqrt{3}d\beta + 2\sqrt{6Dd}\eta)}{\sqrt{2b_2/q}} + \frac{\sigma_2}{\sqrt{b_1}}.$$

Theorem 3. *Suppose $\{x_t, y_t, z_t\}_{t=0}^{T-1}$ be generated from Algorithm 1 by using the **UniGE** zeroth-order gradient estimator, and let $\alpha_t = \frac{1}{t+1}$, $\theta_t = \frac{1}{(t+1)(t+2)}$, $\gamma_t = (1+\theta_t)\eta_t$, $\eta = \eta_t = T^{-\frac{1}{2}}$, $\beta = d^{-1}T^{-\frac{1}{2}}$, $b_2 = q$, and $b_1 = T/d$, then we have*

$$\mathbb{E}[\mathcal{G}(z_\zeta)] = \frac{1}{T}\sum_{t=1}^{T-1} \mathbb{E}[\mathcal{G}(z_t)] \leq O(\frac{\sqrt{d}}{T^{\frac{1}{2}}}) + O(\frac{\sqrt{d}\ln(T)}{T^{\frac{3}{2}}}),$$

where z_ζ is chosen uniformly randomly from $\{z_t\}_{t=0}^{T-1}$.

Remark 3. *Theorem 3 shows that the Acc-SZOFW (UniGE) algorithm has $O(\sqrt{d}T^{-\frac{1}{2}})$ convergence rate. When $\mod(t,q) = 0$, the Acc-SZOFW (UniGE) algorithm needs b_1 samples to estimate the zeroth-order gradient v_t at each iteration and needs T/q iterations, otherwise it needs $2b_2$ samples to estimate v_t at each iteration and needs T iterations. By $\sqrt{d}T^{-\frac{1}{2}} \leq \epsilon$, we choose $T = d\epsilon^{-2}$, and let $b_2 = q = \epsilon^{-1}$ and $b_1 = \epsilon^{-2}$, the Acc-SZOFW has the function query complexity of $b_1 T/q + 2b_2 T = O(d\epsilon^{-3})$ for finding an ϵ-stationary point.*

5.2. Convergence Properties of Acc-SZOFW* Algorithm

In this subsection, we study the convergence properties of the Acc-SZOFW* Algorithm based on the CooGE and UniGE, respectively. The detailed proofs are provided in the Appendix A.2.

5.2.1. ACC-SZOFW* (COOGE) ALGORITHM

Lemma 3. *Suppose the zeroth-order gradient* $v_t = \hat{\nabla}_{coo} f_{\xi_t}(z_t) + (1-\rho_t)\big(v_{t-1} - \hat{\nabla}_{coo} f_{\xi_t}(z_{t-1})\big)$ *be generated from Algorithm 2. Let* $\alpha_t = \frac{1}{t+1}$, $\theta_t = \frac{1}{(t+1)(t+2)}$, $\gamma_t = (1+\theta_t)\eta_t$, $\eta = \eta_t \leq (t+1)^{-a}$ *and* $\rho_t = t^{-a}$ *for some* $a \in (0,1]$ *and the smoothing parameter* $\mu = \mu_t \leq d^{-\frac{1}{2}}(t+1)^{-a}$, *then we have*

$$\mathbb{E}\|v_t - \nabla f(z_t)\| \leq L\sqrt{d}\mu + \sqrt{C}(t+1)^{-\frac{a}{2}}, \qquad (7)$$

where $C = \frac{2(12L^2D^2+12L^2+3\sigma_1^2)}{2-2^{-a}-a}$ *for some* $a \in (0,1]$.

Theorem 4. *Suppose* $\{x_t, y_t, z_t\}_{t=0}^{T-1}$ *be generated from Algorithm 2 by using the* ***CooGE*** *zeroth-order gradient estimator. Let* $\alpha_t = \frac{1}{t+1}$, $\theta_t = \frac{1}{(t+1)(t+2)}$, $\eta = \eta_t = T^{-\frac{2}{3}}$, $\gamma_t = (1+\theta_t)\eta_t$, $\rho_t = t^{-\frac{2}{3}}$ *for* $t \geq 1$ *and* $\mu = d^{-\frac{1}{2}}T^{-\frac{2}{3}}$, *then we have*

$$\mathbb{E}[\mathcal{G}(z_\zeta)] = \frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}[\mathcal{G}(z_t)] \leq O(\frac{1}{T^{\frac{1}{3}}}) + O(\frac{\ln(T)}{T^{\frac{4}{3}}}),$$

where z_ζ *is chosen uniformly randomly from* $\{z_t\}_{t=0}^{T-1}$.

Remark 4. *Theorem 4 shows that the Acc-SZOFW*(CooGE) algorithm has* $O(T^{-\frac{1}{3}})$ *convergence rate. It needs* $2d$ *samples to estimate the zeroth-order gradient* v_t *at each iteration, and needs* T *iterations. For finding an* ϵ*-stationary point,* i.e., *ensuring* $\mathbb{E}[\mathcal{G}(z_\zeta)] \leq \epsilon$, *by* $T^{-\frac{1}{3}} \leq \epsilon$, *we choose* $T = \epsilon^{-3}$. *Thus the Acc-SZOFW* has the function query complexity of* $2dT = O(d\epsilon^{-3})$. ***Note that*** *the Acc-SZOFW* algorithm only requires a small mini-batch size such as* 2 *and reaches the same function query complexity as the Acc-SZOFW algorithm that requires large batch sizes* $b_2 = \epsilon^{-1}$ *and* $b_1 = \epsilon^{-2}$. *For clarity, we need to emphasize that the* ***mini-batch size*** *denotes the sample size required at each iteration, while the* ***query-size*** *(in Table 1) denotes the function query size required in estimating one zeroth-order gradient in these algorithms. In fact, there exists a positive correlation between them. For example, in the Acc-SZOFW* algorithm, the mini-batch size is 2, and the corresponding query-size is* $2d$.

5.2.2. ACC-SZOFW* (UNIGE) ALGORITHM

Lemma 4. *Suppose the zeroth-order gradient* $v_t = \hat{\nabla}_{uni} f_{\xi_t}(z_t) + (1-\rho_t)\big(v_{t-1} - \hat{\nabla}_{uni} f_{\xi_t}(z_{t-1})\big)$ *be generated from Algorithm 2. Let* $\alpha_t = \frac{1}{t+1}$, $\theta_t = \frac{1}{(t+1)(t+2)}$, $\gamma_t = (1+\theta_t)\eta_t$, $\eta = \eta_t \leq (t+1)^{-a}$ *and* $\rho_t = t^{-a}$ *for some* $a \in (0,1]$ *and the smoothing parameter* $\beta = \beta_t \leq d^{-1}(t+1)^{-a}$, *then we have*

$$\mathbb{E}\|v_t - \nabla f(z_t)\| \leq \frac{\beta L d}{2} + \sqrt{C}(t+1)^{-\frac{a}{2}}, \qquad (8)$$

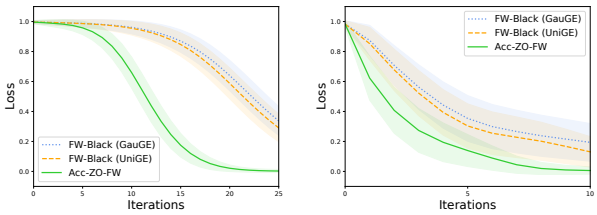
where $C = \frac{24dL^2D^2+3L^2+2\sigma_2^2}{2-2^{-a}-a}$ *for some* $a \in (0,1]$.

Theorem 5. *Suppose* $\{x_t, y_t, z_t\}_{t=0}^{T-1}$ *be generated from Algorithm 2 by using the* ***UniGE*** *zeroth-order gradient estimator. Let* $\alpha_t = \frac{1}{t+1}$, $\theta_t = \frac{1}{(t+1)(t+2)}$, $\eta = \eta_t = T^{-\frac{2}{3}}$, $\gamma_t = (1+\theta_t)\eta_t$, $\rho_t = t^{-\frac{2}{3}}$ *for* $t \geq 1$ *and* $\beta = d^{-1}T^{-\frac{2}{3}}$, *then we have*

$$\mathbb{E}[\mathcal{G}(z_\zeta)] = \frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}[\mathcal{G}(z_t)] \leq O(\frac{\sqrt{d}}{T^{\frac{1}{3}}}) + O(\frac{\sqrt{d}\ln(T)}{T^{\frac{4}{3}}}),$$

where z_ζ *is chosen uniformly randomly from* $\{z_t\}_{t=0}^{T-1}$.

Remark 5. *Theorem 5 states that the Acc-SZOFW*(UniGE) algorithm has* $O(\sqrt{d}T^{-\frac{1}{3}})$ *convergence rate. It needs* 2 *samples to estimate the zeroth-order gradient* v_t *at each iteration, and needs* T *iterations. By* $\sqrt{d}T^{-\frac{1}{3}} \leq \epsilon$, *we choose* $T = d^{\frac{3}{2}}\epsilon^{-3}$. *Thus, the Acc-SZOFW* has the function query complexity of* $2T = O(d^{\frac{3}{2}}\epsilon^{-3})$ *for finding an* ϵ*-stationary point.*

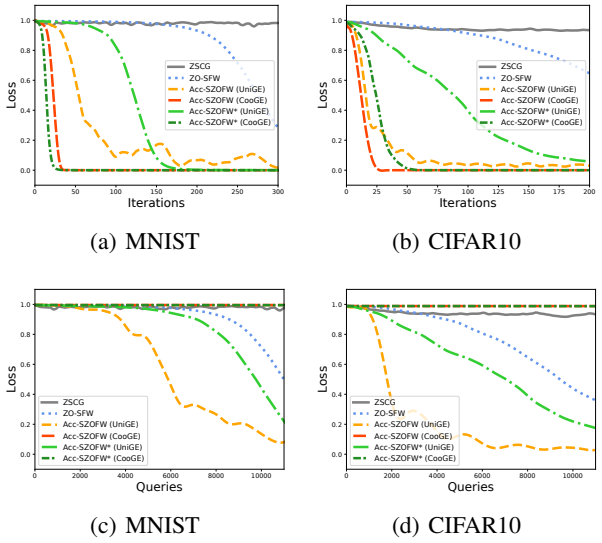


(a) MNIST (b) CIFAR10

Figure 1. The convergence of attack loss against iterations of three algorithms on the SAP problem.

6. Experiments

In this section, we evaluate the performance of our proposed algorithms on two applications: 1) generating adversarial examples from black-box deep neural networks (DNNs) and 2) robust black-box binary classification with ℓ_1 norm bound constraint. In the first application, we focus on two types of black-box adversarial attacks: *single adversarial perturbation* (SAP) against an image and *universal adversarial perturbation* (UAP) against multiple images. Specifically, we apply the SAP to demonstrate the efficiency of our deterministic Acc-ZO-FW algorithm and compare with the FW-Black (Chen et al., 2018) algorithm. While we apply the UAP and robust black-box binary classification to verify the efficiency of our stochastic algorithms (i.e., Acc-SZOFW and Acc-SZOFW*) and compare with the ZO-SFW (Sahu et al., 2019) algorithm and the ZSCG (Balasubramanian & Ghadimi, 2018) algorithm. All of our experiments are conducted on a server with an Intel Xeon 2.60GHz CPU and an NVIDIA Titan Xp GPU. Our implementation is based on PyTorch and the code to reproduce our results is publicly available at https://github.com/TLMichael/Acc-SZOFW.



(a) MNIST (b) CIFAR10

(c) MNIST (d) CIFAR10

Figure 2. Comparison of six algorithms for the UAP problem. Above: the convergence of attack loss against iterations. Below: the convergence of attack loss against queries.

6.1. Black-box Adversarial Attack

In this subsection, we apply the zeroth-order algorithms to generate adversarial perturbations to attack the pre-trained black-box DNNs, whose parameters are hidden and only its outputs are accessible. Let (a, b) denote an image a with its true label $b \in \{1, 2, \cdots, K\}$, where K is the total number of image classes. For the SAP, we will design a perturbation x for a single image (a, b); For the UAP, we will design a universal perturbation x for multiple images $\{a_i, b_i\}_{i=1}^n$. Following (Guo et al., 2019), we solve the untargeted attack problem as follows:

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} p(b_i \mid a_i + x), \quad \text{s.t. } \|x\|_\infty \leq \varepsilon \tag{9}$$

where $p(\cdot \mid a)$ represents probability associated with each class, that is, the final output after softmax of neural network. In the problem (9), we normalize the pixel values to $[0, 1]^d$.

In the experiment, we use the pre-trained DNN models on MNIST (LeCun et al., 2010) and CIFAR10 (Krizhevsky et al., 2009) datasets as the target black-box models, which can attain 99.16% and 93.07% test accuracy, respectively. In the SAP experiment, we choose $\varepsilon = 0.3$ for MNIST and $\varepsilon = 0.1$ for CIFAR10. In the UAP experiment, we choose $\varepsilon = 0.3$ for both MNIST dataset and CIFAR10 dataset. For fair comparison, we choose the mini-batch size $b = 20$ for all stochastic zeroth-order methods. We refer readers to Appendix A.3 for more details of the experimental setups and the generated adversarial examples by our proposed algorithms.

Figure 1 shows that the convergence behaviors of three algorithms on SAP problem, where for each curve, we generate 1000 adversarial perturbations on MNIST and 100 adversarial perturbations on CIFAR10, the mean value of loss are plotted and the range of standard deviation is shown as a shadow overlay. For both datasets, the results show that the attack loss values of our Acc-ZO-FW algorithm faster decrease than those of the FW-Black algorithms, as the iteration increases, which demonstrates the superiority of our novel momentum technique and CooGE used in the Acc-ZO-FW algorithm.

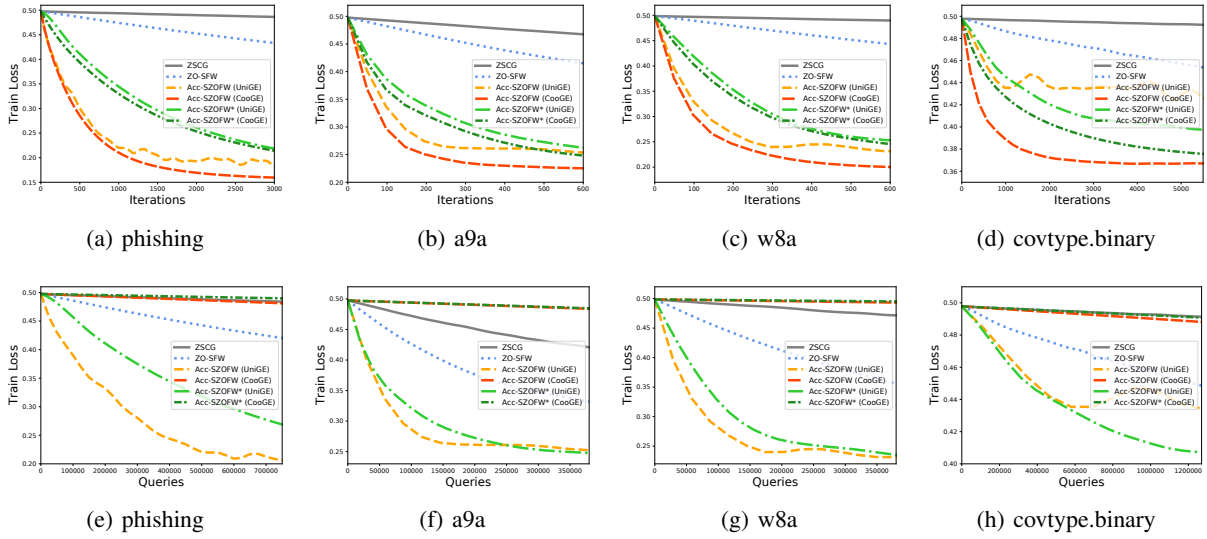
Figure 2 shows that the convergence of six algorithms on UAP problem. For both datesets, the results show that all of our accelerated zeroth-order algorithms have faster convergence speeds (i.e. less iteration complexity) than the existing algorithms, while the Acc-SZOFW (UniGE) algorithm and the Acc-SZOFW* (UniGE) have faster convergence speeds (i.e. less function query complexity) than other algorithms (especially ZSCG and ZO-SFW), which verifies that the effectiveness of the variance reduced technique and the novel momentum technique in our accelerated algorithms. We notice that the periodic jitter of the curve of Acc-SZOFW (UniGE), which is due to the gradient variance reduction period of the variance reduced technique and the imprecise estimation of the uniform smoothing gradient estimator makes the jitter more significant. The jitter is less obvious in Acc-SZOFW (CooGE). Figure 2(c) and Figure 2(d) represent the attack loss against the number of function queries. We observe that the performance of our CooGE-based algorithms degrade since the need of large number of queries to construct coordinate-wise gradient estimates. From these results, we also find that the CooGE-based methods can not be competent to high-dimensional datasets due to estimating each coordinate-wise gradient required at least d queries. In addition, the performance of the Acc-SZOFW algorithms is better than the Acc-SZOFW* algorithms in most cases, which is due to the considerable mini-batch size used in the Acc-SZOFW algorithms.

6.2. Robust Black-box Binary Classification

In this subsection, we apply the proposed algorithms to solve the robust black-box binary classification task. Given a set of training samples $(a_i, l_i)_{i=1}^n$, where $a_i \in \mathbb{R}^d$ and $l_i \in \{-1, +1\}$, we find optimal parameter $x \in \mathbb{R}^d$ by solving the problem:

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(x), \quad \text{s.t. } \|x\|_1 \leq \theta, \tag{10}$$

where $f_i(x)$ is the black-box loss function, that only returns the function value given an input. Here, we specify the loss function $f_i(x) = \frac{\sigma^2}{2}(1 - \exp(-\frac{(l_i - a_i^T x)^2}{\sigma^2}))$, which is the *nonconvex* robust correntropy induced loss. In the



(a) phishing (b) a9a (c) w8a (d) covtype.binary

(e) phishing (f) a9a (g) w8a (h) covtype.binary

Figure 3. Comparison of six algorithms for robust black-box binary classification. Above: the convergence of train loss against iterations. Below: the convergence of train loss against queries.

experiment, we use four public real datasets[1]. We set $\sigma = 10$ and $\theta = 10$. For fair comparison, we choose the mini-batch size $b = 100$ for all stochastic zeroth-order methods. In the experiment, we use four public real datasets, which are summarized in Table 2. For each dataset, we use half of the samples as training data and the rest as testing data. We elaborate the details of the parameter setting in Appendix A.4.

Table 2. Real datasets for black-box binary classification.

DATA SET	#SAMPLES	#FEATURES	#CLASSES
phishing	11,055	68	2
a9a	32,561	123	2
w8a	49,749	300	2
covtype.binary	581,012	54	2

Figure 3 shows that the convergence of six algorithms on the black-box binary classification problem. We see that the results are similar as in the case of the UAP problem. For all datasets, the results show that all of our accelerated algorithms have faster convergence speeds (i.e. less iteration complexity) than the existing algorithms, while the Acc-SZOFW (UniGE) algorithm and the Acc-SZOFW* (UniGE) have faster convergence speeds (i.e. less function query complexity) than other algorithms (especially ZSCG and ZO-SFW), which further demonstrates the efficiency of our accelerated algorithms. Similar to Figure 2, the periodic jitter of the curve of Acc-SZOFW (UniGE) also appears and seems to be more intense in the covtype.binary dataset. We speculate that this is because the variance of the random gradient estimator is too high in this situation. We also provides the convergence of test loss in Appendix A.4, which is analogous to those of train loss.

[1]These data are from the website https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

7. Conclusions

In the paper, we proposed a class of accelerated stochastic gradient-free and projection-free (zeroth-order Frank-Wolfe) methods. In particular, we also proposed a momentum accelerated framework for the Frank-Wolfe methods. Specifically, we presented an accelerated stochastic zeroth-order Frank-Wolfe (Acc-SZOFW) method based on the variance reduced technique of SPIDER and the proposed momentum accelerated technique. Further, we proposed a novel accelerated stochastic zeroth-order Frank-Wolfe (Acc-SZOFW*) to relax the large mini-batch size required in the Acc-SZOFW. Moreover, both the Acc-SZOFW and Acc-SZOFW* methods obtain a lower query complexity, which improves the state-of-the-art query complexity in both finite-sum and stochastic settings.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. This paper was partially supported by the Natural Science Foundation of China (NSFC) under Grant No. 61806093 and No. 61682281, and the Key Program of NSFC under Grant No. 61732006, and Jiangsu Postdoctoral Research Grant Program No. 2018K004A.

References

Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

Allen-Zhu, Z. and Hazan, E. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pp. 699–707, 2016.

Balasubramanian, K. and Ghadimi, S. Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. In *Advances in Neural Information Processing Systems*, pp. 3455–3464, 2018.

Chen, J., Zhou, D., Yi, J., and Gu, Q. A frank-wolfe framework for efficient and effective adversarial attacks. *arXiv preprint arXiv:1811.10828*, 2018.

Chen, S., Luo, L., Yang, J., Gong, C., Li, J., and Huang, H. Curvilinear distance metric learning. In *Advances in Neural Information Processing Systems*, pp. 4223–4232, 2019a.

Chen, X., Liu, S., Xu, K., Li, X., Lin, X., Hong, M., and Cox, D. Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization. In *Advances in Neural Information Processing Systems*, pp. 7202–7213, 2019b.

Cutkosky, A. and Orabona, F. Momentum-based variance reduction in non-convex sgd. In *Advances in Neural Information Processing Systems*, pp. 15210–15219, 2019.

Defazio, A., Bach, F., 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*, pp. 1646–1654, 2014.

Duchi, J. C., Jordan, M. I., Wainwright, M. J., and Wibisono, A. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pp. 689–699, 2018.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2): 95–110, 1956.

Gao, X., Jiang, B., and Zhang, S. On the information-adaptive variants of the admm: an iteration complexity perspective. *Journal of Scientific Computing*, 76(1):327–363, 2018.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Ghadimi, S. and Lan, G. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

Ghadimi, S., Lan, G., and Zhang, H. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155 (1-2):267–305, 2016.

Guo, C., Gardner, J., You, Y., Wilson, A. G., and Weinberger, K. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pp. 2484–2493, 2019.

Hassani, H., Karbasi, A., Mokhtari, A., and Shen, Z. Stochastic conditional gradient++. *arXiv preprint arXiv:1902.06992*, 2019.

Hazan, E. and Kale, S. Projection-free online learning. *arXiv preprint arXiv:1206.4657*, 2012.

Hazan, E. and Luo, H. Variance-reduced and projection-free stochastic optimization. In *ICML*, pp. 1263–1271, 2016.

Huang, F., Gao, S., Chen, S., and Huang, H. Zeroth-order stochastic alternating direction method of multipliers for nonconvex nonsmooth optimization. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 2549–2555. AAAI Press, 2019a.

Huang, F., Gao, S., Pei, J., and Huang, H. Nonconvex zeroth-order stochastic admm methods with lower function query complexity. *arXiv preprint arXiv:1907.13463*, 2019b.

Huang, F., Gu, B., Huo, Z., Chen, S., and Huang, H. Faster gradient-free proximal stochastic methods for nonconvex nonsmooth optimization. In *AAAI*, pp. 1503–1510, 2019c.

Huang, F., Gao, S., Pei, J., and Huang, H. Momentum-based policy gradient methods. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3996–4007, 2020.

Iusem, A. On the convergence properties of the projected gradient method for convex optimization. *Computational & Applied Mathematics*, 22(1):37–52, 2003.

Jaggi, M. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML*, pp. 427–435, 2013.

Ji, K., Wang, Z., Zhou, Y., and Liang, Y. Improved zeroth-order variance reduced algorithms and analysis for non-convex optimization. In *International Conference on Machine Learning*, pp. 3100–3109, 2019.

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

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lacoste-Julien, S. Convergence rate of frank-wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

Lacoste-Julien, S. and Jaggi, M. On the global linear convergence of frank-wolfe optimization variants. In *NeurIPS*, pp. 496–504, 2015.

Lacoste-Julien, S., Jaggi, M., Schmidt, M., and Pletscher, P. Block-coordinate frank-wolfe optimization for structural svms. In *ICML*, pp. 53–61, 2013.

Lan, G. and Zhou, Y. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

Lei, L., Ju, C., Chen, J., and Jordan, M. I. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, pp. 2348–2358, 2017.

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

Liu, S., Chen, J., Chen, P.-Y., and Hero, A. Zeroth-order online alternating direction method of multipliers: Convergence analysis and applications. In *The Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84, pp. 288–297, 2018a.

Liu, S., Kailkhura, B., Chen, P.-Y., Ting, P., Chang, S., and Amini, L. Zeroth-order stochastic variance reduction for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pp. 3727–3737, 2018b.

Liu, S., Li, X., Chen, P.-Y., Haupt, J., and Amini, L. Zeroth-order stochastic projected gradient descent for nonconvex optimization. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1179–1183. IEEE, 2018c.

Malik, D., Pananjady, A., Bhatia, K., Khamaru, K., Bartlett, P., and Wainwright, M. Derivative-free methods for policy optimization: Guarantees for linear quadratic systems. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2916–2925, 2019.

Nesterov, Y. *Introductory Lectures on Convex Programming Volume I: Basic course*. Kluwer, Boston, 2004.

Nesterov, Y. and Spokoiny, V. G. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takáč, M. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2613–2621. JMLR. org, 2017a.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takáč, M. Stochastic recursive gradient algorithm for nonconvex optimization. *arXiv preprint arXiv:1705.07261*, 2017b.

Nitanda, A. Stochastic proximal gradient descent with acceleration techniques. In *Advances in Neural Information Processing Systems*, pp. 1574–1582, 2014.

Qu, C., Li, Y., and Xu, H. Non-convex conditional gradient sliding. In *ICML*, pp. 4205–4214, 2018.

Reddi, S. J., Hefny, A., Sra, S., Poczos, B., and Smola, A. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pp. 314–323, 2016.

Roux, N. L., Schmidt, M., and Bach, F. R. A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in neural information processing systems*, pp. 2663–2671, 2012.

Sahu, A. K., Zaheer, M., and Kar, S. Towards gradient free and projection free stochastic optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3468–3477, 2019.

Shen, Z., Fang, C., Zhao, P., Huang, J., and Qian, H. Complexities in projection-free stochastic non-convex minimization. In *AISTATS*, pp. 2868–2876, 2019.

Tran-Dinh, Q., Pham, N. H., Phan, D. T., and Nguyen, L. M. A hybrid stochastic optimization framework for stochastic composite nonconvex optimization. *arXiv preprint arXiv:1907.03793*, 2019.

Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Wang, Z., Ji, K., Zhou, Y., Liang, Y., and Tarokh, V. Spiderboost: A class of faster variance-reduced algorithms for nonconvex optimization. *arXiv preprint arXiv:1810.10690*, 2018.

Wang, Z., Ji, K., Zhou, Y., Liang, Y., and Tarokh, V. Spiderboost and momentum: Faster variance reduction algorithms. In *Advances in Neural Information Processing Systems*, pp. 2403–2413, 2019.

Xie, J., Shen, Z., Zhang, C., Qian, H., and Wang, B. Stochastic recursive gradient-based methods for projection-free online learning. *arXiv preprint arXiv:1910.09396*, 2019.

Xu, Y. and Yang, T. Frank-wolfe method is automatically adaptive to error bound condition. *arXiv preprint arXiv:1810.04765*, 2018.

Yurtsever, A., Sra, S., and Cevher, V. Conditional gradient methods via stochastic path-integrated differential estimator. In *ICML*, pp. 7282–7291, 2019.

Zhang, M., Shen, Z., Mokhtari, A., Hassani, H., and Karbasi, A. One sample stochastic frank-wolfe. *arXiv preprint arXiv:1910.04322*, 2019.

Zhou, D., Xu, P., and Gu, Q. Stochastic nested variance reduction for nonconvex optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 3925–3936. Curran Associates Inc., 2018.