~~Projection-free~~ Constrained Stochastic Nonconvex Optimization with State-dependent Markov Data

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study a projection-free conditional gradient-type algorithm for constrained nonconvex stochastic optimization problems with Markovian data. In particular, we focus on the case when the transition kernel of the Markov chain is state-dependent. Such stochastic optimization problems arise in various machine learning problems including strategic classification and reinforcement learning. For this problem, we establish that the number of calls to the stochastic first-order oracle and the linear minimization oracle to obtain an appropriately defined ϵ-stationary point, are of the order $\mathcal{O}(1/\epsilon^{2.5})$ and $\mathcal{O}(1/\epsilon^{5.5})$ respectively. We also empirically demonstrate the performance of our algorithm on the problem of strategic classification with neural networks.

1 Introduction

We consider the following stochastic optimization problem

$$\operatorname*{argmin}_{\theta\in\Theta} f(\theta) = \operatorname*{argmin}_{\theta\in\Theta} \mathbb{E}\left[F(\theta; x)\right], \tag{1}$$

where (i) the expectation is taken over the stationary distribution, π_θ, of the random vector x, (ii) F (and hence f) is a potentially non-convex function in θ, and (iii) Θ is a compact and convex constraint set. Stochastic approximation algorithms for solving problem (1), given an independent and identically distributed (`iid`) data stream $\{x_k\}_k$ drawn from π, are well-studied. Such `iid` assumptions are commonly made in various machine learning and statistical problems including empirical risk minimization [SSBD14], sparse recovery [BJMO12] and compressed sensing [FR13, Lan20]. We refer to [MB11, ABRW12, RSS12, GL13, SZ13, LZ16, ACD+19] for a partial list of non-asymptotic upper and lower bounds on the oracle complexity of widely-used stochastic approximation algorithms like the Stochastic Gradient Descent (SGD) and the Stochastic Conditional Gradient Algorithm.

Our focus in this work is on the case when the data sequence $\{x_k\}_k$ is drawn from a Markov chain with a state-dependent transition kernel P_θ. Such a setting arises in several machine learning applications including but not limited to strategic classification [HMPW16, CDP15, MDPZH20, LW22] and reinforcement learning [Bar92, GSK13, ZJM21, KMMW19, QW20]. Despite their prevalence in practice, a deeper understanding of the non-asymptotic oracle complexity of stochastic approximation for Markovian data is only now starting to emerge. We establish non-asymptotic oracle complexity results for the stochastic conditional gradient algorithm for non-convex constrained stochastic optimization with Markovian data. To establish our results, from a methodological point-of-view, we leverage the moving-average stochastic gradient estimation technique recently used in [ZSM+20, GRW20, XBG22] in the context of constrained optimization with `iid` data. This technique avoids having to a use a mini-batch of samples in each iteration, which turns out to be crucial in the non-`iid` setup we consider. From a theoretical point-of-view, we assume the so-called drift conditions, a classical assumption in Markov Chain literature [AMP05]. This ensures the existence

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

of a solution to the Poisson equation associated with the underlying Markov chain [DMPS18] which enables one to decompose the noise present in the stochastic gradient into three components: a martingale difference sequence, a time-decaying sequence, and a telescopic sum type sequence. The key idea of our paper is to use this decomposition to construct an auxiliary sequence of iterates with a time-decaying noise-variance and show that these sequence of iterates are *close* to the iterates of the original sequence produced by our algorithm. This novel technique in then used in combination with a merit-function based analysis to establish the oracle complexity results.

1.1 Motivating Example

Problems of the form in (1) arise in various important applications, e.g., strategic classification, and reinforcement learning as mentioned above. Below we illustrate the motivation of this work through the example of strategic classification with adapted best response [LW22]. In strategic classification, there is a *learner* whose task is to classify a given dataset which is collected from a set of *agents*. Given the knowledge of the classifier, the agents can distort some of their personal features, in order to get classified in a predetermined target class. This scenario arises in various applications, e.g., spam email filtering, and credit score classification. Optimizing the classifier to classify such strategically modified data where the agents modify the data iteratively can be formulated as problem (1).

Formally, let the classifier be $h(x, \theta)$ where $x \in \mathbb{R}^d$ is the feature and θ is the parameter to be optimized. $h(x; \cdot) : \Theta \to \mathbb{R}$ is potentially nonconvex. Let the loss function be logistic loss which for a sample (x, y), where $y \in \{-1, 1\}$ denotes the corresponding class, is given by,

$$L(\theta; x, y) = \log\left(1 + \exp\left(-h(x; \theta)\right)\right) + (1 - y)h(x; \theta)/2. \tag{2}$$

We use x_S, and x_{-S} to denote the subset of feature x which are respectively strategically modifiable, and non-modifiable by the agents. Then the modified feature (the best response) x'_S reported by the agent is the solution to the following optimization problem:

$$x'_S = \underset{x_S}{\operatorname{argmax}} \left(h(x; \theta) - c(x_S, x'_S)\right), \tag{3}$$

where $c(x, x')$ is the cost of modifying x_S to x'_S. Let the agents iteratively learn x'_S similar to [LW22]. Note that unlike [LW22], where the authors deploy a logistic regression classifier and the closed form solution of the best response is readily known to the agents, it may not be the case in general. In that case the agents have to possibly learn the best response x'_S using some iterative optimization algorithm. For example, if the agents use Gradient Ascent then, at every iteration k, a set $\mathcal{I}_k$ of $n_1 \leq M$ randomly chosen agents out of M agents modify their features as:

$$x^k_{S,i} = \begin{cases} x^{k-1}_{S,i} + \alpha\left(\nabla h(x^{k-1}_{S,i}; \theta_k) - \nabla c(x^{k-1}_{S,i}, x^0_{S,i})\right) & i \in \mathcal{I}_k \\ x^{k-1}_{S,i} & i \notin \mathcal{I}_k \end{cases} \tag{4}$$

where α is the stepsize. With a little abuse of notation, we use $\nabla h(x^{k-1}_{S,i}; \theta)$ in (4) to denote the fact that the gradient is with respect to $x^{k-1}_{S,i}$ while $x_{-S,i}$ remains unchanged. This introduces the state-dependent Markov chain dynamics in the training data. The objective function, analogous to $f(\theta)$ in (1), is

$$\min_{\theta \in \Theta} \mathbb{E}_{\pi_\theta}\left[L(\theta; x, y)\right],$$

where π_θ is the stationary joint distribution of (x, y), and Θ is a convex and compact set, e.g., sparsity inducing constraint $\|\theta\|_1 \leq R$ from some $R > 0$. The loss evaluated at a single data point (x, y), $L(\theta; x, y)$, is analogous to $F(\theta; x)$ in (1). [DX20], and [LW22] study this problem theoretically and empirically respectively in an unconstrained strongly convex setting. Our results takes a step towards analyzing this problem in constrained nonconvex setting. We empirically show the performance of the stochastic conditional gradient algorithm on a strategic classification problem in Section 4.

1.2 Preliminaries and Main Contributions

Before we present our main contributions, we introduce our convergence criterion. In constrained optimization literature, most commonly used convergence criteria are: (i) *Gradient Mapping* (GM), and (ii) *Frank-Wolfe Gap* (FW-gap). The *Gradient Mapping* at a point $\bar{\theta} \in \Theta$ is defined as

$$\mathcal{G}_\Theta(\bar{\theta}, \nabla f(\bar{\theta}), \beta) \coloneqq \beta\left(\bar{\theta} - \Pi_\Theta\left(\bar{\theta} - \frac{1}{\beta}\nabla f(\bar{\theta})\right)\right), \tag{5}$$

where $\Pi_\Theta(x)$ denotes the orthogonal projection of the vector x onto the set Θ, i.e.,

$$\Pi_\Theta\left(\bar{\theta} - \frac{1}{\beta}\nabla f(\bar{\theta})\right) = \underset{y\in\Theta}{\operatorname{argmin}}\left\{\langle \nabla f(\bar{\theta}), y - \bar{\theta}\rangle + \frac{\beta}{2}\|y-\theta\|_2^2\right\}.$$

We will use $\Pi_\Theta(\bar{\theta}, \nabla f(\bar{\theta}), \beta)$ to denote $\Pi_\Theta\left(\bar{\theta} - \nabla f(\bar{\theta})/\beta\right)$ when there is no confusion. Note that when $\Theta \equiv \mathbb{R}^d$ we have $\mathcal{G}_\Theta(\bar{\theta}, \nabla f(\bar{\theta}), \beta) = \nabla f(\bar{\theta})$. In other words, for constrained optimization gradient mapping plays an analogous role of the gradient for unconstrained optimization. The gradient mapping is a frequently used measure in the literature as a convergence criterion for nonconvex constrained optimization [Nes18]. We should emphasize here that although the gradient mapping cannot be computed in the stochastic setting, one can still use it as a convergence measure.

[BG22] shows that the above notion of convergence criterion is closely related to the so-called *Frank-Wolfe Gap*. The FW-gap is defined as

$$g_\Theta(\bar{\theta}, \nabla f(\bar{\theta})) := \min_{y\in\Theta} \langle \nabla f(\bar{\theta}), y - \bar{\theta}\rangle. \tag{6}$$

The following proposition from [BG22] establishes the relation between the gradient mapping criterion and the Frank-Wolfe gap:

Proposition 1.1 [BG22] *Let $g_\Theta(\cdot)$ be the Frank-Wolfe gap defined in (6) and $\mathcal{G}_\Theta(\cdot)$ be the gradient mapping defined in (5). Then, we have*

$$\|\mathcal{G}_\Theta(\bar{\theta}, \nabla f(\bar{\theta}), \beta)\|^2 \le g_\Theta(\bar{\theta}, \nabla f(\bar{\theta})), \qquad \forall \bar{\theta} \in \Theta.$$

Moreover, under standard regularity assumption in smooth optimization (specifically, Assumption 2.1, and 2.2), we have

$$g_\Theta(\bar{\theta}, \nabla f(\bar{\theta})) \le L\left\|\mathcal{G}_\Theta(\bar{\theta}, \nabla f(\bar{\theta}), \beta)\right\|_2/\beta. \tag{7}$$

In this work we use a suboptimality measure, closely related to both GM and the FW-gap. At point $\bar{\theta} \in \Theta$, we define the suboptimality measure $V(\bar{\theta}, z) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ as [GRW20]

$$V(\bar{\theta}, z) := \left\|\Pi_\Theta\left(\bar{\theta} - z/\beta\right) - \bar{\theta}\right\|_2^2 + \left\|z - \nabla f(\bar{\theta})\right\|_2^2, \tag{8}$$

where z, formally defined in Algorithm 1, is the moving-average estimate of $\nabla f(\bar{\theta})$. We show the relation among $V(\theta, z)$, GM $\mathcal{G}_\Theta(\theta, z, \beta)$, and FW-gap $g(\theta_k, \nabla f(\theta_k))$ in the following proposition.

Proposition 1.2 *Let $\{z_k\}$ be the sequence generated in Algorithm 1. Then, for $k = 1, 2, \cdots, N$,*

$$\max\{\|\mathcal{G}_\Theta(\theta_k, z_k, \beta)\|_2^2, g(\theta_k, \nabla f(\theta_k))\} \le V(\theta_k, z_k).$$

The proof is provided in Appendix A. Observe that by establishing ϵ-stationarity in terms of $\mathbb{E}\left[V(\theta_k, z_k)\right]$ one can have a tighter bound on FW-gap than gradient mapping ϵ-stationarity as described in Definition 1. Indeed $\mathbb{E}\left[V(\theta_k, z_k)\right] \le \epsilon$ implies $\mathbb{E}\left[g_\Theta(\theta_k, \nabla f(\theta_k))\right] \le \epsilon$ unlike gradient mapping criterion where, according to (7), $\mathbb{E}\left[\|\mathcal{G}(\theta_k, z_k)\|_2^2\right] \le \epsilon$ implies $\mathbb{E}\left[g_\Theta(\theta_k, \nabla f(\theta_k))\right] \le \sqrt{\epsilon}$.

The main objective of this work is to find an ϵ-stationary solution to (1), where an ϵ-stationary solution is defined as follows:

Definition 1 *A point $\bar{\theta}$ is said to be an ϵ-stationary solution to (1), if $\mathbb{E}\left[V(\bar{\theta}, z)\right] \le \epsilon$, where the expectation is taken over all the randomness involved in the problem.*

For stochastic Frank-Wolfe-type algorithms, the oracle complexity is measured in terms of number of calls to the Stochastic First-order Oracle (SFO) and the Linear Minimization Oracle (LMO) used to the solve the sub-problems of the algorithm which involves minimizing a linear function over the convex constraint set. Formally, we have the following definition.

Definition 2 *For a given point $\theta \in \Theta$, SFO returns the stochastic gradient $\nabla F(\theta, x)$. Given a vector z, LMO returns a vector $v := \operatorname{argmin}_{y\in\Theta}\langle z, y\rangle$.*

Hence, in this work, the oracle complexity is measured in terms of the number of calls to SFO and LMO required by the proposed algorithm to obtain an ϵ-stationary solution as in Definition 1. With the above preliminaries, we now list our **main contributions**:

		iid		non-iid			
				State-independent MC		State-dependent MC	
Algorithm	Criterion	SFO	LMO	SFO	LMO	SFO	LMO
1-SFW [ZSM+20]	FW-gap	$\mathcal{O}\left(\epsilon^{-3}\right)$	$\mathcal{O}\left(\epsilon^{-3}\right)$	✗	✗	✗	✗
(ASA+ICG) [XBG22]	GM	$\mathcal{O}\left(\epsilon^{-2}\right)$	$\mathcal{O}\left(\epsilon^{-3}\right)$	✗	✗	✗	✗
(ASA+ICG) [This paper]	GM/FW-Gap	✗	✗	$\tilde{\mathcal{O}}\left(\epsilon^{-2}\right)$	$\tilde{\mathcal{O}}\left(\epsilon^{-3}\right)$	$\mathcal{O}\left(\epsilon^{-2.5}\right)$	$\mathcal{O}\left(\epsilon^{-5.5}\right)$

Table 1: Oracle complexity of projection-free one-sample stochastic conditional gradient algorithms for constrained non-convex optimization, to find an ϵ-stationary point.

- In Theorem 3.1, we show that the number of calls to the SFO and LMO required by the stochastic conditional gradient-type method in Algorithm 1, with *state-dependent* Markovian data, is of order $\mathcal{O}(\epsilon^{-2.5})$ and $\mathcal{O}(\epsilon^{-5.5})$ respectively. To the best of our knowledge, these are the first oracle complexity results for projection-free one-sample stochastic optimization algorithm for constrained nonconvex optimization in the Markovian setting.
- In Theorem 3.2, for the sake of completion, we also show that the number of calls to the SFO and LMO required for the case of *state-independent* Markovian data is of the order $\tilde{\mathcal{O}}(\epsilon^{-2})$ and $\tilde{\mathcal{O}}(\epsilon^{-3})$ respectively. In particular, this turns out to be of the same order as that of `iid` data ignoring the logarithmic factors.

A summary of the our contributions is provided in Table 1. We also empirically evaluate our algorithm on a strategic classification problem with 2-layer neural network classifier and show that the proposed method obtains encouraging results. We provide an experiment on single-index model regression with sparsity-inducing nuclear-norm ball constraint in Appendix C.

1.3 Related Work

Stochastic Optimization with Dependent Data. Understanding stochastic approximation algorithms like SGD with dependent data in the asymptotic setting has been well-explored in the optimization literature. We refer to [KY03, Bor09, BMP12] for a text-book introduction to such classical results. A few recent results include [AMP05, TD17]. In the unconstrained non-asymptotic setting, [DAJJ12] studies convex optimization with ergodic data sequence. [DL22] uses multi-level gradient estimator and analyze AdaGrad for nonconvex optimization with Markovian Data. Block coordinate descent with homogeneous Markov chain has been analyzed in [SSXY20] for nonconvex unconstrained optimization. [DX20] studies stochastic optimization with decision-dependent data distribution for strongly convex functions in the context of strategic classification.

Sample-average approximation algorithms for constrained convex optimization with ϕ-mixing data was considered in [WPT+21]. [SSY18], and [AL22] analyze projected SGD for constrained non-convex optimization with time-homogeneous Markov chain. None of these works consider state-dependent data distribution except [DX20]. But unlike [DX20], we consider constrained nonconvex optimization. There also exists work in the reinforcement learning literature on understanding stochastic optimization with Markovian data; see, for example [XXLZ20, BRS18, DNPR20]. However, such works are invariably focused on specific objective functions arising in the reinforcement learning setup, while our focus is on obtaining results for a general class of functions.

Conditional Gradient-Type Method. There has been significant recent advancements in the conditional gradient algorithm literature although it was developed long back [FW56, LP66]; see [Mig94, Jag13, LJJ15, LJJ15, HJN15, GKS21, BS17], for a non-exhaustive list of recent works. [HK12, HL16] provided expected oracle complexity results for stochastic conditional gradient algorithm in the stochastic convex setup. Better rates were provided by a sliding procedure in [LZ16]. In the non-convex setting, [RSPS16, YSC19, HL16] considered variance reduced stochastic conditional gradient algorithms, and provided expected oracle complexities. [QLX18] analyzed the sliding algorithm in the non-convex setting and provided results for the gradient mapping criterion. All of the above works use increasing orders of mini-batch based gradient-estimate.

To avoid mini-batches, a moving-average gradient estimator based on only one-sample in each iteration for a stochastic conditional gradient-type algorithm was proposed in [MHK20] and [ZSM+20] for the convex and non-convex setting. However, several restrictive assumptions have been made in [MHK20] and [ZSM+20]. Specifically, [ZSM+20] requires that the stochastic gradient $G_1(x, \xi_1)$ has uniformly bounded function value, gradient-norm, and Hessian spectral-norm, and the distribution of the random vector ξ_1 has an absolutely continuous density p such that the norm of the gradient of

$\log p$ and spectral norm of the Hessian of $\log p$ has finite fourth and second-moments respectively. In contrasts, we do not require such stringent assumptions.

2 Assumptions

We now introduce the precise assumptions we make in this work. Let $\mathcal{F}_k$ be the filtration generated by $\{\theta_0, \cdots, \theta_k, z_0, \cdots, z_k, x_1, \cdots, x_k\}$. For any mapping $g : \mathbb{R}^d \to \mathbb{R}^d$ define the following norm with respect to a function $\mathcal{V} : \mathbb{R}^d \to [1, \infty)$:

$$\|g\|_{\mathcal{V}} = \sup_{x \in \mathscr{X}} \left(\|g(x)\|_2 / \mathcal{V}(x) \right),$$

and let $L_{\mathcal{V}} = \{g : \mathbb{R}^d \to \mathbb{R}^d, \sup_{x \in \mathscr{X}} \|g\|_{\mathcal{V}} < \infty\}$.

Assumption 2.1 (Constraint set) *The set $\Theta \subset \mathbb{R}^d$ is convex and closed with $\max_{x,y \in \Theta} \|x - y\|_2 \leq D_\Theta$, form some $D_\Theta > 0$.*

Assumption 2.2 *Let f be a continuously differentiable function.*

Assumption 2.3 *Let $\xi_{k+1}(\theta_k, x_{k+1}) \coloneqq \nabla F(\theta_k, x_{k+1}) - \nabla f(\theta_k)$. Then,*

$$\mathbb{E}\left[\|\xi_{k+1}(\theta_k, x_{k+1})\|_2^2 \,|\mathcal{F}_k\right] \leq \sigma_1^2 \qquad \mathbb{E}\left[\|\nabla F(\theta_k, x_{k+1})\|_2^2 \,|\mathcal{F}_k\right] \leq \sigma_2^2 \qquad \sigma^2 \coloneqq \max(\sigma_1^2, \sigma_2^2).$$

Assumption 2.4 *Let $\{x_k\}_k$ be a Markov chain with transition kernel P_θ. For any $\theta \in \Theta$, P_θ is irreducible and aperiodic. Additionally, there exists a function $\mathcal{V} : \mathbb{R}^d \to [1, \infty)$ and a constant $\alpha \geq 2$ such that for any compact set $\Theta' \subset \Theta$:*

(a) There exist a set $C \subset \mathbb{R}^d$, an integer I, constants $0 < \lambda < 1$, b, κ, $\delta > 0$, and a probability measure ν such that,

$$\sup_{\theta \in \Theta'} P_\theta^l \mathcal{V}^\alpha(x) \leq \lambda \mathcal{V}^\alpha(x) + bI(x \in C) \quad \forall x \in \mathbb{R}^d, \tag{9}$$

$$\sup_{\theta \in \Theta'} P_\theta \mathcal{V}^\alpha(x) \leq \kappa \mathcal{V}^\alpha(x) \quad \forall x \in \mathbb{R}^d, \tag{10}$$

$$\inf_{\theta \in \Theta'} P_\theta^l(x, A) \geq \delta \nu(A) \quad \forall x \in C, \forall A \in \mathcal{B}_{\mathbb{R}^d}. \tag{11}$$

where $\mathcal{B}_{\mathbb{R}^d}$ is the Borel σ-algebra over $\mathbb{R}^d$.

(b) There exists a constant $c > 0$, such that, for all $x \in \mathbb{R}^d$,

$$\sup_{\theta \in \Theta'} \|\nabla F(\theta, x)\|_{\mathcal{V}} \leq c, \tag{12}$$

$$\sup_{(\theta, \theta') \in \Theta'} \|\nabla F(\theta, x) - \nabla F(\theta', x)\|_{\mathcal{V}} \leq c \|\theta - \theta'\|_2. \tag{13}$$

(c) There exists a constant $c > 0$, such that, for all $(\theta, \theta') \in \Theta' \times \Theta'$,

$$\|P_\theta g - P_{\theta'} g\|_{\mathcal{V}} \leq c \|g\|_{\mathcal{V}} \|\theta - \theta'\|_2 \quad \forall g \in L_{\mathcal{V}} \tag{14}$$

$$\|P_\theta g - P_{\theta'} g\|_{\mathcal{V}^\alpha} \leq c \|g\|_{\mathcal{V}^\alpha} \|\theta - \theta'\|_2 \quad \forall g \in L_{\mathcal{V}^\alpha}. \tag{15}$$

Some comments regarding the assumptions are in order. Assumption 2.1, and Assumption 2.2 are common for constrained optimization [GRW20, XBG22, AL22, ZSM+20]. Assumption 2.1, and Assumption 2.2 together imply the Lipschitz continuity of $f(\cdot)$, i.e., there is a constant $L > 0$ such that for any $\theta_1, \theta_2 \in \Theta$, we have $|f(\theta_1) - f(\theta_2)| \leq L \|\theta_1 - \theta_2\|_2$. Assumption 2.3 is common in stochastic optimization literature. Assumption 2.4(a) is a frequently used assumption in Markov chain literature. It implies that for every $\theta \in \Theta$, there exists a stationary distribution $\pi_\theta(x)$, and the chain is $\mathcal{V}^\alpha$-uniformly ergodic [AMP05]. Assumption 2.4(c) provides smoothness guarantee on the function $f(\cdot)$. More formally, we have the following proposition.

Proposition 2.1 (Lipschitz continuous gradient [AMP05]) *Let Assumption 2.4 be true. Then $f(\cdot)$ has Lipschitz continuous gradient, i.e., there is a constant $L_G > 0$ such that for any $\theta_1, \theta_2 \in \Theta$:*

$$\|\nabla f(\theta_1) - \nabla f(\theta_2)\|_2 \leq L_G \|\theta_1 - \theta_2\|_2. \tag{16}$$

Finally, the most important implication of Assumption 2.4 is that it ensures the existence and regularity of a solution $u(\theta, x)$ to Poisson equation of the transition kernel P_θ given by $u(\theta, x) - P_\theta u(\theta, x) = \nabla F(\theta, x) - \nabla f(\theta)$. Solution of Poisson equation has been crucial in analyzing additive functionals of Markov chain (see [AMP05] for details). In this work, the Poisson equation solution facilitates a decomposition of the noise as presented in Lemma 3.1 which is a key component of our analysis.

3 Main Result

In this section we present our main result on the oracle complexity to establish a bound on $\mathbb{E}\left[V(\theta_k, z_k)\right]$. In order to do so we use Algorithm 1, and 2 similar to [XBG22]. If an exact

Algorithm 1 Inexact Averaged Stochastic Approximation (I-ASA)

Input: $z_0, \theta_0 \in \mathbb{R}^d$, $\eta_k = (N+k)^{-a}$, $1/2 < a < 1$, β.

for $k = 1, 2, \cdots, N$ **do**

$$y_k = \begin{cases} \min\limits_{y \in \Theta} \left\{ \langle z_k, y - \theta_k \rangle + \frac{\beta}{2} \|y - \theta_k\|_2^2 \right\} & \text{(Projection)} \\ \text{ICG}(z_k, \theta_k, \beta, t_k, \omega) & \text{(No Projection)} \end{cases}$$

$\theta_{k+1} = \theta_k + \eta_{k+1}(y_k - \theta_k)$

$z_{k+1} = (1 - \eta_{k+1}) z_k + \eta_{k+1} \nabla F(\theta_k, x_{k+1})$

end for

Output: θ_k

Algorithm 2 Inexact Conditional Gradient (ICG)

Input: $z, \theta, \beta, t, \omega$.

Set $w_0 = \theta$

for $i = 1, 2, \cdots, t-1$ **do**

Find v_i such that

$$\langle v_i, z + \beta(w_i - \theta) \rangle \leq \operatorname*{argmin}_{v \in \Theta} \langle v, z + \beta(w_i - \theta) \rangle + \beta \omega \mathcal{D}_\Theta^2 / (i+2)$$

$w_{i+1} = (1 - \mu_i) w_i + \mu_i v_i$ where $\mu_i = \frac{2}{i+2}$

end for

Output: w_t

minimizer of the following subproblem, which is the projection of $\theta_k - z_k/\beta$ on to Θ, is available, then Algorithm 1 is same as `ASA` algorithm introduced in [GRW20].

$$\min_{y \in \Theta} \left\{ \langle z_k, y - \theta_k \rangle + \frac{\beta}{2} \|y - \theta_k\|_2^2 \right\}. \tag{17}$$

When a projection operator is unavailable or computationally costly, we use Algorithm 2 instead to solve (17). At iteration k, Algorithm 2 finds an approximate solution to (17) based on the conditional gradient algorithm. Algorithm 2 needs access to LMO which is often much cheaper and simpler to compute than projection operator. We should emphasize that our results are not limited to `ICG` method but are valid for any method which can solve (17) within an error of the order of $\{\eta_k\}$.

Theorem 3.1 *Let Assumption 2.1-2.4 be true. Then, for Algorithm 1,*

(a) when a projection operator is available, choosing

$$\eta_k = (N+k)^{-3/5}, \quad \beta = 1 \tag{18}$$

for $k = 1, 2, \cdots, N$ we have

$$\mathbb{E}\left[V(\theta_R, z_R)\right] = \mathcal{O}\left(N^{-\frac{2}{5}}\right),$$

(b) when Algorithm 2 is used to solve (17)*,choosing*

$$\eta_k = (N+k)^{-3/5}, \quad t_k = \eta_k^{-2}, \quad \beta = 1, \quad \omega = 1, \quad \mu_i = 2/(i+2) \tag{19}$$

for $k = 1, 2, \cdots, N$ we have

$$\mathbb{E}\left[V(\theta_R, z_R)\right] = \mathcal{O}\left(N^{-\frac{2}{5}}\right),$$

where the expectations are taken with respect to all the randomness of the algorithm, and an independent integer random variable $R \in \{1, 2, \cdots, N\}$ with probability mass function,

$$P\left(R = k\right) = \eta_k / \sum_{k=1}^{N} \eta_k \quad k \in \{1, 2, \cdots, N\}.$$

Remark 1 *Note that total number of LMO calls are $\sum_{k=1}^{N} t_k = \sum_{k=1}^{N} t_k = \sum_{k=1}^{N} (N+k)^{2a} = \mathcal{O}(N^{11/5})$. In other words, to achieve $\|\mathcal{G}_\Theta(\theta_R, \nabla f(\theta_R), \beta)\|_2^2 \leq \mathbb{E}\left[V(\theta_R, z_R)\right] \leq \epsilon$, SFO and LMO complexities are respectively $\epsilon^{-2.5}$, and $\epsilon^{-5.5}$. Note that the SFO complexity will be $\epsilon^{-2.5}$ as long as one has an approximation of the projection operator with approximation error $\mathcal{O}(\eta_k)$.*

Remark 2 *In Theorem 3.1, one obtains sublinear rate $\max(N^{a-1}, N^{2-4a})$ with $\eta_k = (N+k)^{-a}$ for $1/2 < a < 1$. Choosing $a = 3/5$ provides the fastest rate of convergence.*

Before sketching the outline of the proof, we present the following lemma which provides a decomposition of the noise $\xi_k(\theta_{k-1}, x_k)$ – one of the key result used in the proof of the main theorem. The lemma and its proof are almost same as Lemma A.5 in [Lia10] with the only difference that unlike [Lia10], where the iterates are of SGD, we need to prove it for the iterates of Algorithm 1. We provide the proof in Appendix A.

Lemma 3.1 *Let Assumption 2.1-2.4 be true. Then the following decomposition takes place:*

$$\xi_k(\theta_{k-1}, x_k) = e_k + \nu_k + \zeta_k,$$

where, $\{e_k\}$ is martingale difference sequence, $\mathbb{E}\left[\|\nu_k\|_2\right] \leq \eta_k$, and $\zeta_k = (\tilde{\zeta}_k - \tilde{\zeta}_{k+1})/\eta_k$, where $\mathbb{E}\left[\|\tilde{\zeta}_k\|_2\right] \leq \eta_k$.

Outline of the proof: A key step in the analysis of Algorithm 1 involves controlling the expectation of interaction with noise of the form $\langle \nabla f(\theta_k) - \nabla f(\theta_{k-1}), \xi_{k+1}(\theta_k, x_{k+1})\rangle$. For `iid` or martingale difference data it is easy to control because $\mathbb{E}\left[\langle \nabla f(\theta_k) - \nabla f(\theta_{k-1}), \xi_{k+1}(\theta_k, x_{k+1})\rangle | \mathcal{F}_k\right] = 0$. But this is no longer true for Markov chain data. To resolve the issue, first notice that under our assumptions, the noise sequence ξ_k can be decomposed into the sum of a martingale difference sequence $\{e_k\}$ and some residual terms $\{\nu_k\}$, and $\{\zeta_k\}$ as shown in Lemma 3.1. Then the key step is to introduce a different sequence of hypothetical iterates $(\tilde{\theta}_k, \tilde{y}_k, \tilde{z}_k)$ for which the noise is small enough so that we can bound $\mathbb{E}\left[V(\tilde{\theta}_k, \tilde{z}_k)\right]$, and then show that these hypothetical iterates and the original sequence generated by Algorithm 1 are close enough so that $\mathbb{E}\left[V(\theta_k, z_k)\right]$ is of the same order as $\mathbb{E}\left[V(\tilde{\theta}_k, \tilde{z}_k)\right]$. This step is the main novelty of the proof.

Specifically, consider the following sequence:

$$\tilde{\theta}_0 = \theta_0 \quad \tilde{z}_0 = z_0 \tag{20}$$

$$\tilde{y}_k = \underset{y \in \Theta}{\operatorname{argmin}} \left\{ \left\langle \tilde{z}_k, y - \tilde{\theta}_k \right\rangle + \frac{\beta}{2} \left\| y - \tilde{\theta}_k \right\|_2^2 \right\} \tag{21}$$

$$\tilde{\theta}_{k+1} = \tilde{\theta}_k + \eta_{k+1}(\tilde{y}_k - \tilde{\theta}_k) \tag{22}$$

$$\tilde{z}_{k+1} = z_{k+1} + \tilde{\zeta}_{k+2} \tag{23}$$

This also means,

$$\tilde{z}_{k+1} = (1 - a\eta_{k+1})\tilde{z}_k + a\eta_{k+1}\left(\nabla f(\theta_k) + \tilde{\epsilon}_{k+1}\right), \tag{24}$$

where, $\tilde{\epsilon}_k = e_k + \nu_k + \tilde{\zeta}_k$. Note that by Lemma 3.1, $\mathbb{E}\left[e_k\right] = 0$, and $\mathbb{E}\left[\left\|\nu_k + \tilde{\zeta}_k\right\|_2\right] \leq \eta_k$. First we show that by choosing $\eta_k = (N+k)^{-a}$, $1/2 < a < 1$, and $t_k = 1/\eta_k^2$ one has $\mathbb{E}\left[\|\tilde{\theta}_k - \theta_k\|_2^2\right] = \mathcal{O}\left(N^{2-4a}\right)$, and $\mathbb{E}\left[V(\theta_k, z_k)\right] \leq 2\mathbb{E}\left[V(\tilde{\theta}_k, \tilde{z}_k)\right] + \mathcal{O}\left(N^{2-4a}\right)$. Then we establish the bound on $V(\tilde{\theta}_k, \tilde{z}_k)$. Combining the above two facts proves Theorem 3.1. We defer the detailed proof to Appendix A.1.

3.1 State-independent Markov Chain

While our main goal in this work is to analyze Algorithm 1 for constrained nonconvex optimization with state-dependent Markov chain data, we provide the following result on the complexity of Algorithm 1 for Markov chain data with state-independent transition kernel for the sake of completion. Here we use P to denote the transition kernel (as opposed to P_θ for state-dependent kernel). Note that under Assumption 2.4(a), for each θ, the chain is $\mathcal{V}$-uniformly ergodic, and hence, exponentially mixing [MT12] in the following sense:

Definition 3 *A Markov chain is said to be exponentially mixing, if there exists $C, r > 0$ such that, for any initial state x,*

$$\|P^n(x,\cdot) - \pi\|_{\mathcal{V}} \leq C\exp(-rn), \tag{25}$$

where $P^n(x,\cdot)$ is the distribution of X_n with initial state $X_0 = x$.

Now we present our result on the complexity of Algorithm 1 to find an ϵ-stationary solution to (1) for exponentially-mixing Markov chain data with state-independent transition kernel.

Theorem 3.2 *Let Assumption 2.1-2.3 be true. Let Assumption 2.4(a)-(b) be true with P_θ replaced by P. Then, for Algorithm 1,*

(a) when the projection operator is available, choosing

$$\eta_k = 1/\sqrt{N}, \quad \beta = 1 \tag{26}$$

for $k = 1, 2, \cdots, N$ we have

$$\mathbb{E}\left[V(\theta_R, z_R)\right] = \mathcal{O}\left(\log N/\sqrt{N}\right),$$

(b) when Algorithm 2 is used, choosing

$$\eta_k = 1/\sqrt{N}, \quad t_k = \lceil\sqrt{k}\rceil, \quad \beta = 1, \quad \omega = 1, \quad \mu_i = 2/(i+2) \tag{27}$$

for $k = 1, 2, \cdots, N$ we have

$$\mathbb{E}\left[V(\theta_R, z_R)\right] = \mathcal{O}\left(\log N/\sqrt{N}\right),$$

where the expectation is taken with respect to all the randomness of the algorithm, and an independent integer random variable $R \in \{1, 2, \cdots, N\}$ whose probability mass function is given by,

$$P\left(R = k\right) = \eta_k / \sum_{k=1}^{N} \eta_k \quad k \in \{1, 2, \cdots, N\}.$$

We defer the proof to the Appendix.

Remark 3 *To find an ϵ-stationary point, the total number of calls to SFO and LMO are $\tilde{\mathcal{O}}\left(\epsilon^{-2}\right)$, and $\tilde{\mathcal{O}}\left(\epsilon^{-3}\right)$, where $\tilde{\mathcal{O}}(\cdot)$ denotes the order ignoring logarithmic factors.*

Remark 4 *The authors of [AL22] obtain the same rate as in Theorem 3.2 for constrained (but projection-based) nonconvex optimization with state-independent exponentially mixing data. In the state-dependent case, since the transition kernel of the Markov chain is controlled by θ_k, and the transition kernel is assumed to be only Lipschitz smooth in θ (15), the chain does not necessarily exponentially mix. In the state-independent case, since the chain mixes exponentially we obtain the same rate as well. While their results are for projection-based algorithms, we analyze a projection-free LMO-based algorithm since LMO is often computationally cheaper than projection.*

4 Experimental Evaluation

In this section we illustrate our algorithm on the strategic classification problem as described in Section 1.1 with the `GiveMeSomeCredit`[1] dataset. The main task is a credit score classification

[1] Available at `https://www.kaggle.com/c/GiveMeSomeCredit/data`

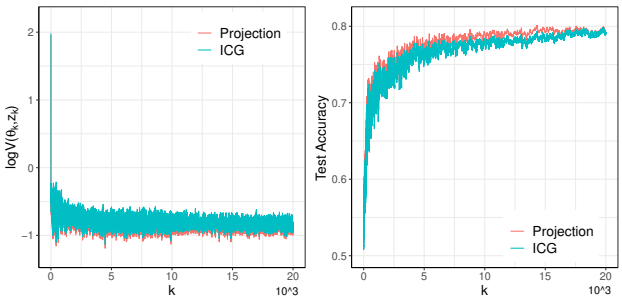

Figure 1: Strategic Classification: (*Left*): Performance of Algorithm 1 with and without the projection operator. (*Right*): Test Accuracy with Algorithm 1 with and without the projection operator.

problem where the bank (learner) has to decide whether a loan should be granted to a client. Given the knowledge of the classifier the clients (agents) can distort some of their personal traits in order to get approved for a loan. Here we use a 2-layer neural network with width m as the classifier, given by

$$h(x; \mathcal{W}, \mathcal{A}, \mathcal{B}) = \sum_{i=1}^{m} \mathcal{A}_i \upsilon(\mathcal{W}_i^\top x + \mathcal{B}_i),$$

where $\upsilon(\cdot)$ is the activation function, $\mathcal{W}_i \in \mathbb{R}^d$, $\mathcal{W} = [\mathcal{W}_1, \mathcal{W}_2, \cdots, \mathcal{W}_d]^\top \in \mathbb{R}^{m \times d}$, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_m) \in \mathbb{R}^m$, $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \cdots, \mathcal{B}_m) \in \mathbb{R}^m$. We will use θ to collectively denote $(\mathcal{W}, \mathcal{A}, \mathcal{B})$. We impose the constraint of sparsity on the classifier given by $\|\theta\|_1 \leq R$ for some $R > 0$. As loss function we consider logistic loss as shown in (2). We consider a quadratic cost given by $c(x, x') = \|x_S - x'_S\|_2^2 / (2\lambda)$ where λ is the sensitivity of the underlying distribution on θ. We assume that the agents iteratively learn x'_S similar to [LW22]. Note that unlike [LW22], the closed form of best response is not known here. So we assume that the agents use Gradient Ascent (GA) to learn the best response. For $\|\theta\|_1 \leq R$ constraint, the LMO in Algorithm 2 at iteration k is given by $-R \operatorname{sign}(q_i)$, where $i = \operatorname{argmax}_{j=1,\cdots,d} |q_j|$, $q = z + \beta(w_k - \theta)$, and q_j is the j-th coordinate of q. We select a subset of randomly chosen $M = 2000$ samples (agents) such that the dataset is balanced. Each agent has 10 features. Note that since Algorithm 1 computes the gradient on one sample at every iterate, the computation time is independent of the total number of agents. We assume that the agents can modify Revolving Utilization, Number of Open Credit Lines, and Number of Real Estate Loans or Lines. In this experiment we set $n_1 = 200$. Similar to [LW22], we set $\alpha = 0.5\lambda$, and $\lambda = 0.01$. For the classifier, the activation function is chosen as *sigmoidal*, and $m = 400$. We set $N = 20000$, and $R = 4000$. All the parameters of Algorithm 1 are chosen as described in (19). Figure 4 shows that Algorithm 1 finds an ϵ-stationary point of the strategic classification problem. We show that Algorithm 1 performs comparably with Averaged Stochastic Approximation with the projection operator. Each curve in Figure 4 is an average of 50 repetitions.

5 Discussion

In this work we provide oracle complexity results for the stochastic conditional gradient algorithm to find an ϵ-stationary point of a constrained nonconvex optimization problem with state-dependent Markovian data. In Theorem 3.1, we show that the number of calls to the SFO and LMO required by the stochastic conditional gradient-type method in Algorithm 1, with *state-dependent* Markovian data, is $\mathcal{O}(\epsilon^{-2.5})$ and $\mathcal{O}(\epsilon^{-5.5})$ respectively. To the best of our knowledge, these are the first oracle complexity results in this setting. In Theorem 3.2, we show that SFO and LMO complexity in the case of state-independent Markovian data is $\tilde{\mathcal{O}}(\epsilon^{-2})$ and $\tilde{\mathcal{O}}(\epsilon^{-3})$ respectively, which matches the corresponding results in the `iid` setting.

There are various avenues for further extensions. Establishing lower bounds on the oracle complexity of projection-free algorithms in the Markovian data setting is extremely interesting. It is also intriguing to establish upper and lower bounds on the oracle complexity for more general types of dependent data sequences arising in applications, including ϕ and α mixing sequences. Yet another exciting direction is that of designing algorithms adaptive to the dependency in the data that achieve potentially better oracle complexity bounds.

References

[ABRW12] Alekh Agarwal, Peter Bartlett, Pradeep Ravikumar, and Martin Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, 2012. (Cited on page 1.)

[ACD+19] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019. (Cited on page 1.)

[AL22] Ahmet Alacaoglu and Hanbaek Lyu. Convergence and complexity of stochastic subgradient methods with dependent data for nonconvex optimization. *arXiv preprint arXiv:2203.15797*, 2022. (Cited on pages 4, 5, and 8.)

[AMP05] Christophe Andrieu, Éric Moulines, and Pierre Priouret. Stability of stochastic approximation under verifiable conditions. *SIAM Journal on control and optimization*, 44(1):283–312, 2005. (Cited on pages 1, 4, 5, 6, 15, and 24.)

[Bar92] Peter L Bartlett. Learning with a slowly changing distribution. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 243–252, 1992. (Cited on page 1.)

[BG22] Krishnakumar Balasubramanian and Saeed Ghadimi. Zeroth-order nonconvex stochastic optimization: Handling constraints, high dimensionality, and saddle points. *Foundations of Computational Mathematics*, 22(1):35–76, 2022. (Cited on page 3.)

[BJMO12] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012. (Cited on page 1.)

[BMP12] Albert Benveniste, Michel Métivier, and Pierre Priouret. *Adaptive algorithms and stochastic approximations*, volume 22. Springer Science & Business Media, 2012. (Cited on page 4.)

[Bor09] Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009. (Cited on page 4.)

[BRS18] Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on learning theory*, pages 1691–1692. PMLR, 2018. (Cited on page 4.)

[BS17] Amir Beck and Shimrit Shtern. Linearly convergent away-step conditional gradient for non-strongly convex functions. *Mathematical Programming*, 164(1-2):1–27, 2017. (Cited on page 4.)

[CDP15] Yang Cai, Constantinos Daskalakis, and Christos Papadimitriou. Optimum statistical estimation with strategic data sources. In *Conference on Learning Theory*, pages 280–296. PMLR, 2015. (Cited on page 1.)

[CXY21] Rong Chen, Han Xiao, and Dan Yang. Autoregressive models for matrix-valued time series. *Journal of Econometrics*, 222(1):539–560, 2021. (Cited on page 27.)

[DAJJ12] John C Duchi, Alekh Agarwal, Mikael Johansson, and Michael I Jordan. Ergodic mirror descent. *SIAM Journal on Optimization*, 22(4):1549–1578, 2012. (Cited on page 4.)

[DL22] Ron Dorfman and Kfir Y Levy. Adapting to mixing time in stochastic optimization with markovian data. *arXiv preprint arXiv:2202.04428*, 2022. (Cited on page 4.)

[DMPS18] Randal Douc, Eric Moulines, Pierre Priouret, and Philippe Soulier. *Markov chains*. Springer, 2018. (Cited on page 2.)

[DNPR20] Thinh T Doan, Lam M Nguyen, Nhan H Pham, and Justin Romberg. Convergence rates of accelerated markov gradient descent with applications in reinforcement learning. *arXiv preprint arXiv:2002.02873*, 2020. (Cited on page 4.)

[DX20] Dmitriy Drusvyatskiy and Lin Xiao. Stochastic optimization with decision-dependent distributions. *arXiv preprint arXiv:2011.11173*, 2020. (Cited on pages 2 and 4.)

[FR13] Simon Foucart and Holger Rauhut. An invitation to compressive sensing. In *A mathematical introduction to compressive sensing*, pages 1–39. Springer, 2013. (Cited on page 1.)

[FW56] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956. (Cited on page 4.)

[GKS21] Dan Garber, Atara Kaplan, and Shoham Sabach. Improved complexities of conditional gradient-type methods with applications to robust matrix recovery problems. *Mathematical Programming*, 186(1):185–208, 2021. (Cited on page 4.)

[GL13] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. (Cited on page 1.)

[GRW20] Saeed Ghadimi, Andrzej Ruszczynski, and Mengdi Wang. A single timescale stochastic approximation method for nested stochastic optimization. *SIAM Journal on Optimization*, 30(1):960–979, 2020. (Cited on pages 1, 3, 5, 6, 15, 22, and 23.)

[GSK13] Yair Goldberg, Rui Song, and Michael R Kosorok. Adaptive q-learning. In *From Probability to Statistics and Back: High-Dimensional Models and Processes–A Festschrift in Honor of Jon A. Wellner*, pages 150–162. Institute of Mathematical Statistics, 2013. (Cited on page 1.)

[HJN15] Zaid Harchaoui, Anatoli Juditsky, and Arkadi Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152(1-2):75–112, 2015. (Cited on pages 4 and 28.)

[HK12] Elad Hazan and Satyen Kale. Projection-free online learning. In *29th International Conference on Machine Learning, ICML 2012*, pages 521–528, 2012. (Cited on page 4.)

[HL16] Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271, 2016. (Cited on page 4.)

[HMPW16] Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122, 2016. (Cited on page 1.)

[Jag13] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pages 427–435. PMLR, 2013. (Cited on pages 4, 16, 24, and 28.)

[JS10] Martin Jaggi and Marek Sulovskỳ. A simple algorithm for nuclear norm regularized problems. In *ICML*, 2010. (Cited on page 28.)

[KMMW19] Belhal Karimi, Blazej Miasojedow, Eric Moulines, and Hoi-To Wai. Non-asymptotic analysis of biased stochastic approximation scheme. In *Conference on Learning Theory*, pages 1944–1974. PMLR, 2019. (Cited on page 1.)

[KY03] Harold Kushner and George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003. (Cited on page 4.)

[Lan20] Guanghui Lan. *First-order and stochastic optimization methods for machine learning*. Springer, 2020. (Cited on page 1.)

[Lia10] Faming Liang. Trajectory averaging for stochastic approximation mcmc algorithms. *The Annals of Statistics*, 38(5):2823–2856, 2010. (Cited on pages 7 and 15.)

[LJJ15] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504, 2015. (Cited on page 4.)

[LP66] Evgeny Levitin and Boris Polyak. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6(5):1–50, 1966. (Cited on page 4.)

[LW22] Qiang Li and Hoi-To Wai. State dependent performative prediction with stochastic approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 3164–3186. PMLR, 2022. (Cited on pages 1, 2, and 9.)

[LZ16] Guanghui Lan and Yi Zhou. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016. (Cited on pages 1 and 4.)

[MB11] Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in neural information processing systems*, 24, 2011. (Cited on page 1.)

[MDPZH20] Celestine Mendler-Dünner, Juan Perdomo, Tijana Zrnic, and Moritz Hardt. Stochastic optimization for performative prediction. *Advances in Neural Information Processing Systems*, 33:4929–4939, 2020. (Cited on page 1.)

[MHK20] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *Journal of machine learning research*, 2020. (Cited on page 4.)

[Mig94] Athanasios Migdalas. A regularization of the frank—wolfe method and unification of certain nonlinear programming methods. *Mathematical Programming*, 65(1):331–345, 1994. (Cited on page 4.)

[MT12] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012. (Cited on page 8.)

[Nes18] Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018. (Cited on page 3.)

[QLX18] Chao Qu, Yan Li, and Huan Xu. Non-convex conditional gradient sliding. In *International Conference on Machine Learning*, pages 4208–4217. PMLR, 2018. (Cited on page 4.)

[QW20] Guannan Qu and Adam Wierman. Finite-time analysis of asynchronous stochastic approximation and q-learning. In *Conference on Learning Theory*, pages 3185–3205. PMLR, 2020. (Cited on page 1.)

[RSPS16] Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic frank-wolfe methods for nonconvex optimization. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1244–1251. IEEE, 2016. (Cited on page 4.)

[RSS12] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1571–1578, 2012. (Cited on page 1.)

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. (Cited on page 1.)

[SSXY20] Tao Sun, Yuejiao Sun, Yangyang Xu, and Wotao Yin. Markov chain block coordinate descent. *Computational Optimization and Applications*, 75(1):35–61, 2020. (Cited on page 4.)

[SSY18] Tao Sun, Yuejiao Sun, and Wotao Yin. On markov chain gradient descent. *Advances in neural information processing systems*, 31, 2018. (Cited on page 4.)

[SZ13] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International conference on machine learning*, pages 71–79. PMLR, 2013. (Cited on page 1.)

[TD17] Vladislav B Tadić and Arnaud Doucet. Asymptotic bias of stochastic gradient search. *The Annals of Applied Probability*, 27(6):3255–3304, 2017. (Cited on page 4.)

[WPT+21] Yafei Wang, Bo Pan, Wei Tu, Peng Liu, Bei Jiang, Chao Gao, Wei Lu, Shangling Jui, and Linglong Kong. Sample average approximation for stochastic optimization with dependent data: Performance guarantees and tractability. *arXiv preprint arXiv:2112.05368*, 2021. (Cited on page 4.)

[XBG22] Tesi Xiao, Krishnakumar Balasubramanian, and Saeed Ghadimi. A projection-free algorithm for constrained stochastic multi-level composition optimization. *arXiv preprint arXiv:2202.04296*, 2022. (Cited on pages 1, 4, 5, and 6.)

[XXLZ20] Huaqing Xiong, Tengyu Xu, Yingbin Liang, and Wei Zhang. Non-asymptotic convergence of adam-type reinforcement learning algorithms under markovian sampling. *arXiv preprint arXiv:2002.06286*, 2020. (Cited on page 4.)

[YSC19] Alp Yurtsever, Suvrit Sra, and Volkan Cevher. Conditional gradient methods via stochastic path-integrated differential estimator. In *International Conference on Machine Learning*, pages 7282–7291. PMLR, 2019. (Cited on page 4.)

[ZJM21] Kelly Zhang, Lucas Janson, and Susan Murphy. Statistical inference with m-estimators on adaptively collected data. *Advances in Neural Information Processing Systems*, 34, 2021. (Cited on page 1.)

[ZSM+20] Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. One-sample Stochastic Frank-Wolfe. In *International Conference on Artificial Intelligence and Statistics*, pages 4012–4023. PMLR, 2020. (Cited on pages 1, 4, and 5.)

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report the average of 50 trails
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] We did not calculate the exact timings. However, our experiments are fairly small-scale ones run on a personal laptop computer, and our main contributions are theoretical.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]