Conditional Accelerated Lazy Stochastic Gradient Descent

Guanghui Lan [*1] **Sebastian Pokutta** [*1] **Yi Zhou** [*1] **Daniel Zink** [*1]

Abstract

In this work we introduce a conditional accelerated lazy stochastic gradient descent algorithm with optimal number of calls to a stochastic first-order oracle and convergence rate $O(\frac{1}{\varepsilon^2})$ improving over the projection-free, *Online Frank-Wolfe* based stochastic gradient descent of (Hazan and Kale, 2012) with convergence rate $O(\frac{1}{\varepsilon^4})$.

1. Introduction

The conditional gradient method (also known as: Frank-Wolfe algorithm) proposed in (Frank and Wolfe, 1956), gained much popularity in recent years due to its simple projection-free scheme and fast practical convergence rates. We consider the basic convex programming (CP) problem

$$f^* := \min_{x \in X} f(x), \tag{1}$$

where $X \subseteq \mathbb{R}^n$ is a closed convex set and $f : X \to \mathbb{R}$ is a smooth convex function such that $\exists L > 0$,

$$\|f'(x) - f'(y)\|_* \leq L\|x - y\|, \quad \forall x, y \in X. \tag{2}$$

The classic conditional gradient (CG) method solves (1) iteratively by minimizing a series of linear approximations of f over the feasible set X. More specifically, given $x_{k-1} \in X$ at the k-th iteration, it updates x_k according to the following steps:

1) Call the first-order (FO) oracle to compute $(f(x_{k-1}), f'(x_{k-1}))$ and set $p_k = f'(x_{k-1})$.

2) Call the linear optimization (LO) oracle to compute

$$y_k \in \operatorname{argmin}_{x \in X} \langle p_k, x \rangle. \tag{3}$$

3) Set $x_k = (1 - \lambda_k)x_{k-1} + \lambda_k y_k$ for some $\lambda_k \in [0, 1]$.

Compared to most other first-order methods, such as e.g., gradient descent algorithms and accelerated gradient algorithms (Nesterov, 1983; 2004), the CG method is computationally cheaper in some cases, since it only requires the solution of a linear optimization subproblem (3) rather than an often costly projection onto the feasible region X.

There has been extensive and fruitful research on the general class of linear-optimization-based convex programming (LCP) methods (which covers the CG method and its variants) and their application in machine learning (e.g., (Ahipasaoglu and Todd, 2013; Bach et al., 2012; Beck and Teboulle, 2004; Cox et al., 2013; Clarkson, 2010; Freund and Grigas, 2013; Hazan, 2008; Harchaoui et al., 2012; Jaggi, 2011; 2013; Jaggi and Sulovský, 2010; Luss and Teboulle, 2013; Shen et al., 2012; Hazan and Kale, 2012; Lan, 2013; Lan and Zhou, 2014; Braun et al., 2016)). It should be noted that even the computational cost for LO oracle to solve the linear optimization subproblem (3) is high for some complex feasible regions. Recently, several approaches have been considered to address this issue. Jaggi demonstrated practical speed up for the CG method by approximately solving (3) in (Jaggi, 2013). Braun, Pokutta, and Zink in (Braun et al., 2016) proposed a class of modified CG methods, namely the lazy conditional gradient (LCG) algorithms, which calls a weak separation oracle rather than solving the linear subproblem (3) in the classical CG method. In fact, the weak separation oracle is computationally more efficient than approximate minimization used in (Jaggi, 2013), at the expense of not providing any guarantee for function value improvement with respect to (3). Furthermore, as shown in (Jaggi, 2013; Lan, 2013), the total number of iterations for the LCP methods to find an ϵ-solution of (1) (i.e., a point $\bar{x} \in X$, s.t. $f(\bar{x}) - f^* \leq \epsilon$) cannot be smaller than $\mathcal{O}(1/\epsilon)$, which is not improvable even when the objective function f is strongly convex. Improved complexity results can only be obtained under stronger assumptions on the LO oracle or the feasible set (see, e.g., (Garber and Hazan, 2013; Lan, 2013)). However, the $\mathcal{O}(1/\epsilon)$ bound does not preclude the existence of more efficient LCP algorithms for solving (1). Lan and Zhou in (Lan and Zhou, 2014) proposed a class of conditional gradient sliding methods (CGS), which significantly improve the complexity bounds in terms of the number of gradient evaluations while maintaining optimal complexity bounds

[*]Equal contribution [1]ISyE, Georgia Institute of Technology, Atlanta, GA. Correspondence to: Daniel Zink <daniel.zink@gatech.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

for the LO oracle calls required by the LCP methods.

Inspired by (Braun et al., 2016) and (Lan and Zhou, 2014), in this paper we focus on a class of modified LCP methods that require only improving solutions for a certain separation problem rather than solving the linear optimization subproblem (3) explicitly through LO oracle calls while simultaneously minimizing the number of gradient evaluations when performing weak separation over the feasible set X. At first these two objectives seem to be incompatible as (Braun et al., 2016) give up the dual guarantee to simplify the oracle, while the dual guarantee of CG iterations is at the core of the analysis in (Lan and Zhou, 2014). We overcome this impasse by carefully modifying both techniques.

It should be mentioned that Hazan and Kale in (Hazan and Kale, 2012) proposed the online Frank-Wolfe (OFW) algorithm, which obtains $\mathcal{O}(1/\epsilon^4)$ rate of convergence for stochastic problems. Indeed, if we consider the objective function $f(x) := \mathbb{E}[F(x,\xi)]$ for stochastic optimization, the OFW method can be applied to solve (1) by viewing the iteratively observed function f_t as the current realization of the true objective function f, i.e., $f_t(\cdot) = F(\cdot,\xi_t)$. *Without re-evaluating* the (sub)gradients at the updated points, the OFW obtains $\mathcal{O}(T^{-1/4})$ bound for any (smooth or non-smooth) objective functions (see Theorem 4.4 in (Hazan and Kale, 2012)), which implies $\mathcal{O}(1/\epsilon^4)$ rate of convergence in terms of the number of (sub)gradient evaluations for stochastic optimization. However, we can show that our proposed algorithm obtains $\mathcal{O}(1/\epsilon)$ (resp., $\mathcal{O}(1/\epsilon^2)$) rate of convergence for smooth (resp., non-smooth) stochastic problems, which is much better than the convergence rate of the OFW method. We would like to stress that the stochastic optimization bound in (Hazan and Kale, 2012, Theorem 4.1) which gives a guarantee of $\mathcal{O}(1/\epsilon^2)$, requires to re-evaluate all gradients at the current iterate and as such the number of gradient evaluations required grows quadratically in t. Moreover, Hazan and Luo (2016) proposed two methods for solving the special case of Problem (1) of the form

$$\min_{x\in X} f(x) = \min_{x\in X} \frac{1}{m}\sum_{i=1}^{m} f_i(x),$$

which allows for a potentially smaller number of SFO evaluations than $\mathcal{O}(1/\varepsilon^2)$, the lower bound for the general problem. The two methods Stochastic Variance-Reduced Frank-Wolfe (SVRF) and Stochastic Variance-Reduced Conditional Gradient Sliding (STORC) are obtained by applying the variance reduction idea of Johnson and Zhang (2013) and Mahdavi et al. (2013) to the CG method and the Stochastic CGS method respectively. Both algorithms however need a certain number of exact (or full) gradient evaluations leading to a potentially undesirable dependence on the number of examples m.

Contributions

Our main contributions can be briefly summarized as follows. We consider stochastic smooth optimization, where we have only access unbiased estimators of the gradients of f via a stochastic first-order (SFO) oracle. By incorporating a modified LCG procedure (Braun et al., 2016) into a modified CGS method (Lan and Zhou, 2014) we obtain a new conditional accelerated lazy stochastic gradient descent algorithm (CALSGD) and we show that the number of calls to the weak separation oracle can be optimally bounded by $\mathcal{O}(1/\epsilon)$, while the optimal bound of $\mathcal{O}(1/\epsilon^2)$ on the total number of calls to the SFO oracle can be maintained. In addition, if the exact gradients of f can be accessed by a FO oracle, the latter bound can be significantly improved to $\mathcal{O}(1/\sqrt{\epsilon})$. In order to achieve the above we will present a modified lazy conditional gradient method, and show that the total number of iterations (or calls to the weak separation oracle) performed by it can be bounded by $\mathcal{O}(1/\epsilon)$ under a stronger termination criterion, i.e., the primal-dual gap function.

We also consider strongly convex and smooth functions and show that without enforcing any stronger assumptions on the weak separation oracle or the feasible set X, the total number of calls to the FO (resp., SFO) oracle can be optimally bounded by $\mathcal{O}(\log 1/\epsilon)$ (resp., $\mathcal{O}(1/\epsilon)$) for variants of the proposed method to solve deterministic (resp., stochastic) strongly convex and smooth problems. Furthermore, we also generalize the proposed algorithms to solve an important class of non-smooth convex programming problems with a saddle point structure. By adaptively approximating the original non-smooth problem via a class of smooth functions, we are able to show that the deterministic version of CALSGD can obtain an ϵ-solution within $\mathcal{O}(1/\epsilon)$ number of linear operator evaluations and $\mathcal{O}(1/\epsilon^2)$ number of calls to the weak separation oracle, respectively. The former bound will increase to $\mathcal{O}(1/\epsilon^2)$ for non-smooth stochastic optimization.

Finally, we demonstrate practical speed ups of CALSGD through preliminary numerical experiments for the video co-localization problem, the structured regression problem and quadratic optimization over the standard spectrahedron; an extensive study is beyond the scope of this paper and left for future work. In all cases we report a substantial improvements in performance.

In the main body of the paper we focus on the stochastic smooth case; several other results and their proofs have been relegated to the Supplementary Material.

1.1. Notation and Terminology

Let $X \subseteq \mathbb{R}^n$ be a convex compact set, and $\|\cdot\|_X$ be the norm associated with the inner product in $\mathbb{R}^n$. For the sake

of simplicity, we often skip the subscript in the norm $\|\cdot\|_X$. We define the diameter of the set X as

$$D_X \equiv D_{X,\|\cdot\|} := \max_{x,y\in X} \|x-y\|. \tag{4}$$

For a given norm $\|\cdot\|$, we denote its conjugate by $\|s\|_* = \max_{\|x\|\le 1}\langle s, x\rangle$. For a linear operator $A:\mathbb{R}^n \to \mathbb{R}^m$, we use $\|A\|$ to denote its operator norm defined as $\|A\| := \max_{\|x\|\le 1}\|Ax\|$. Let $f: X \to \mathbb{R}$ be a convex function, we denote its linear approximation at x by

$$l_f(x;y) := f(x) + \langle f'(x), y-x\rangle. \tag{5}$$

Clearly, if f satisfies (2), then

$$f(y) \le l_f(x;y) + \tfrac{L}{2}\|y-x\|^2, \quad \forall\, x,y\in X. \tag{6}$$

Notice that the constant L in (2) and (6) depends on $\|\cdot\|$. Moreover, we say f is *smooth with curvature* at most C, if

$$f(y) \le l_f(x;y) + \tfrac{C}{2}, \quad \forall\, x,y\in X. \tag{7}$$

It is clear that if X is bounded, we have $C \le LD_X^2$. In the following we also use $\mathbb{R}_{++}$ to denote the set of strictly positive reals.

2. Conditional Accelerated Lazy Stochastic Gradient Descent

We now present a new method for stochastic gradient descent that is based on the stochastic conditional gradient sliding (SCGS) method and the parameter-free lazy conditional gradient (LCG) procedure from Section 2.2, which we refer to as the *Conditional Accelerated Lazy Stochastic Gradient Descent (CALSGD)* method.

We consider the stochastic optimization problem:

$$f^* := \min_{x\in X}\{f(x) = \mathbb{E}_\xi[F(x,\xi)]\}, \tag{8}$$

where $f(x)$ is a smooth convex function satisfying (2).

2.1. The Algorithm

Throughout this section, we assume that there exists a stochastic first-order (SFO) oracle, which for a search point $z_k \in X$ outputs a stochastic gradient $F'(z_k,\xi_k)$, s.t.

$$\mathbb{E}\left[F'(z_k,\xi_k)\right] = f'(z_k), \tag{9}$$

$$\mathbb{E}\left[\|F'(z_k,\xi_k) - f'(z_k)\|_*^2\right] \le \sigma^2. \tag{10}$$

If $\sigma = 0$, the stochastic gradient $F'(z_k,\xi_k)$ is the exact gradient at point z_k, i.e., $F'(z_k,\xi_k) = f'(z_k)$.

Our algorithmic framework is inspired by the SCGS method by (Lan and Zhou, 2014). However, instead of applying the classic CG method to solve the projection subproblem appearing in the accelerated gradient (AG) method, the CALSGD method utilizes a modified parameter-free LCG algorithm (see Section 2.2) to approximately solve the subproblem $\psi(x)$ defined in (16) and skips the computations of the stochastic gradient $F'(z,\xi)$ from time to time when performing weak separation over the feasible region X. The main advantages of our method are that it does not solve a traditional projection problem and achieves the optimal bounds on the number of calls to the SFO and LOsep_X oracles (see Oracle 1 in Subsection 2.2) for solving problem (1)-(8). To the authors' best knowledge, no such algorithms have been developed before in the literature; we present the algorithm below in Algorithm 1.

Algorithm 1 Conditional Accelerated Lazy Stochastic Gradient Descent (CALSGD)

Input: Initial point $x_0 \in X$, iteration limit N, and weak separation oracle accuracy $\alpha \ge 1$.
Let $\beta_k \in \mathbb{R}_{++}$, $\gamma_k \in [0,1]$, and $\eta_k \in \mathbb{R}_+$, $k = 1,2,\ldots$, be given and set $y_0 = x_0$.
for $k = 1,2,\ldots,N$ **do**

$$z_k = (1-\gamma_k)y_{k-1} + \gamma_k x_{k-1}, \tag{11}$$

$$g_k = \tfrac{1}{B_k}\textstyle\sum_{j=1}^{B_k} F'(z_k,\xi_{k,j}), \tag{12}$$

$$x_k = \text{LCG}(g_k, \beta_k, x_{k-1}, \alpha, \eta_k), \tag{13}$$

$$y_k = (1-\gamma_k)y_{k-1} + \gamma_k x_k, \tag{14}$$

where $F'(z_k,\xi_{k,j})$, $j = 1,\ldots,B_k$, are stochastic gradients computed by the SFO at z_k.
end for
Output: y_N.

We hasten to make some observations about the CALSGD method. Firstly, we apply mini-batches to estimate the gradient at point z_k, where the parameter $\{B_k\}$ denotes the batch sizes used to compute g_k. It can be easily seen from (9), (10), and (12) that

$$\mathbb{E}[g_k - f'(z_k)] = 0 \ \text{ and } \ \mathbb{E}[\|g_k - f'(z_k)\|_*^2] \le \tfrac{\sigma^2}{B_k}, \tag{15}$$

and hence g_k is an unbiased estimator of $f'(z_k)$. In fact, letting $S_{B_k} = \sum_{j=1}^{B_k}(F'(z_k,\xi_{k,j}) - f'(z_k))$, from (9) and (10), by induction, we have

$$\begin{aligned}
\mathbb{E}\left[\|S_{B_k}\|_*^2\right] &= \mathbb{E}\left[\|S_{B_k-1} + F'(z_k,\xi_{k,B_k}) - f'(z_k)\|_*^2\right] \\
&= \mathbb{E}\left[\|S_{B_k-1}\|_*^2 + \|F'(z_k,\xi_{k,B_k}) - f'(z_k)\|_*^2\right. \\
&\qquad \left. +2\langle S_{B_k-1}, F'(z_k,\xi_{k,B_k}) - f'(z_k)\rangle\right] \\
&= \mathbb{E}\left[\|S_{B_k-1}\|_*^2\right] \\
&\quad + \mathbb{E}\left[\|F'(z_k,\xi_{k,B_k}) - f'(z_k)\|_*^2\right] \\
&= \textstyle\sum_{j=1}^{B_k}\mathbb{E}\left[\|F'(z_k,\xi_{k,j}) - f'(z_k)\|_*^2\right] \le B_k\sigma^2,
\end{aligned}$$

which together with the fact that $g_k - f'(z_k) = \frac{1}{B_k}\sum_{j=1}^{B_k}[F'(z_k,\xi_{k,j}) - f'(z_k)] = \frac{1}{B_k}S_{B_k}$, implies the second relationship in (15).

Secondly, in view of the SCGS method in (Lan and Zhou, 2014), x_k obtained in (13) should be an approximate solution to the gradient sliding subproblem

$$\min_{x\in X}\left\{\psi_k(x) := \langle g_k, x\rangle + \tfrac{\beta_k}{2}\|x - x_{k-1}\|^2\right\}, \qquad (16)$$

such that for some $\eta_k \geq 0$ we have

$$\langle \psi_k'(x_k), x_k - x\rangle = \langle g_k + \beta_k(x_k - x_{k-1}), x_k - x\rangle \leq \eta_k, \qquad (17)$$

for all $x \in X$. If we solve the subproblem (16) exactly (i.e., $\eta_k = 0$), then CALSGD will reduce to the accelerated stochastic approximation method by (Lan, 2009; 2012). However, by employing the LCG procedure (see Procedure 1 in Subsection 2.2), we only need to use a weak separation oracle, but still maintaining the optimal bounds on stochastic first-order oracle as in (Lan, 2009; 2012; Lan and Zhou, 2014).

Thirdly, observe that the CALSGD method so far is conceptual only as we have not yet specified the LCG procedure and the parameters $\{B_k\}$, $\{\beta_k\}$, $\{\gamma_k\}$, and $\{\eta_k\}$. We will come back to this issue after introducing the LCG procedure and establishing its main convergence properties.

2.2. The Parameter-free Lazy Conditional Gradient Procedure

The classical CG method is a well-known projection-free algorithm, which requires only the solution of a linear optimization subproblem (3) rather than the projection over X per iteration. Therefore, it has computational advantages over many other first-order methods when projection over X being costly. The LCG procedure presented in this subsection, a modification of the vanilla LCG method in (Braun et al., 2016), goes several steps further than CG and even vanilla LCG method. Firstly, it replaces LO oracle by a weaker separation oracle LOsep, which is no harder than linear optimization and often much simpler. Secondly, it uses a stronger termination criterion, the Frank-Wolfe gap (cf. (18)), than vanilla LCG method. Finally, it maintains the same order of convergence rate as the CG and the vanilla LCG method.

We present the LOsep oracle in Oracle 1 below.

Oracle 1 Weak Separation Oracle $\text{LOsep}_P(c, x, \Phi, \alpha)$

Input: $c \in \mathbb{R}^n$ linear objective, $x \in P$ point, $\alpha \geq 1$ accuracy, $\Phi > 0$ objective value;

Output: $y \in P$ vertex with either (1) $c^T(x - y) > \Phi/\alpha$, or (2) $y = \text{argmax}_{y\in P} c^T(x - z) \leq \Phi$.

Observe that the oracle has two output modes. In particular, Oracle 1 first verifies whether there exists an improving point $y \in P$ with the required guarantee and if so it outputs this point, which we refer it as a *positive call*. If no such point exists the oracle certifies this by providing the maximizer y, which then also provides a new duality gap. We refer to this case as a *negative call*. The computational advantages of this oracle are that it can reuse previously seen solutions y if they satisfy the improvement condition and even if LO oracle has to be called, the optimization can be terminated early once the improvement condition is satisfied. Finally, the parameter α allows to only approximately satisfy the improvement condition making separation even easier; in our applications we set the parameter α slightly larger than 1.

We present the LCG procedure based on (Braun et al., 2016) below. We adapted the parameter-free version to remove any dependence on hard to estimate parameters. For any smooth convex function ϕ, we define its *duality gap* as

$$gap_{\phi,X}(x) \equiv gap_\phi(x) := \max_{y\in X} \nabla\phi(x)^T(x - y). \qquad (18)$$

Clearly, by convexity the duality gap is an upper bound on $f(x^*) - f(x)$. Given any accuracy parameter $\eta \geq 0$, the LCG procedure solves $\min_{x\in X}\phi(x)$ approximately with accuracy η, i.e., it outputs a point $\bar{u} \in X$, s.t. $gap_\phi(\bar{u}) \leq \eta$.

Procedure 1 Parameter-free Lazy Conditional Gradients (LCG) procedure

Input: access to gradients of smooth convex function ϕ, $u_1 \in X$ vertex, LOsep_X weak linear separation oracle, accuracy $\alpha \geq 1$, duality gap bound η

Output: $\bar{u} \in X$ with bounded duality gap, i.e., $gap_\phi(\bar{u}) \leq \eta$

1: $\Phi_0 \leftarrow \max_{u\in X} \nabla\phi(u_1)^T(u_1 - u)$
2: **for** $t = 1$ **to** $T - 1$ **do**
3: $v_t \leftarrow \text{LOsep}_X(\nabla\phi(u_t), x_t, \Phi_{t-1}, \alpha)$
4: **if not** $\nabla\phi(u_t)^T(u_t - v_t) > \Phi_{t-1}/\alpha$ **then**
5: **if** $\Phi_{t-1} = \eta$ **then**
6: **return** $\bar{u} = u_t$
7: **end if**
8: **else**
9: $\Phi_t \leftarrow \max\left\{\frac{\Phi_{t-1}}{2}, \eta\right\}$ {Update Φ_t}
10: **end if**
11: $\lambda_t \leftarrow \text{argmin}\ \phi((1 - \lambda_t)u_t + \lambda_t v_t)$
12: $u_{t+1} \leftarrow (1 - \lambda_t)u_t + \lambda_t v_t$
13: **end for**

The LCG procedure is a parameter-free algorithm. Note that while line search can be expensive in general, for our subproblems, function evaluation is very cheap. The algorithm needs only one LO oracle call to estimate the initial functional value gap at Line 1. Alternatively, this

can be also done approximately via binary search with LOsep. The algorithm maintains a sequence, $\{\Phi_t\}$, that provides valid upper bounds for the functional value gap at the current iterate, i.e., $\phi(u_t) - \phi^* \leq 2\Phi_{t-1}$ (see Theorem 5.1 of (Braun et al., 2016)), and it halves the value of Φ_t only when the current oracle call is negative. Finally, our LCG procedure exits at Line 5 whenever LOsep_X returns a negative call and $\Phi_{t-1} = \eta$, which ensures that $gap_\phi(\bar{u}) = \max_{y \in X} \langle \nabla\phi(\bar{u}), \bar{u} - y\rangle \leq \eta$.

Theorem 2.1 below provides a bound for the total number of iterations (or calls to the LOsep_X oracle) that the LCG procedure requires to generate a point $\bar{u} \in X$ with $gap_\phi(\bar{u}) \leq \eta$.

Theorem 2.1. *Procedure 1 returns a point $\bar{u} \in X$ such that the duality gap at point $\bar{u}$ is bounded by η, i.e., $gap_\phi(\bar{u}) \leq \eta$. Furthermore, the total number of iterations T (and hence LOsep_X calls) performed by Procedure 1 is at most*

$$T \leq \begin{cases} \kappa + \frac{8\alpha^2 C_\phi}{\eta} + 2, & \eta < \alpha C_\phi; \\ \kappa + 4\alpha + \frac{4\alpha^2 C_\phi}{\eta} + 2, & \eta \geq \alpha C_\phi, \end{cases} \tag{19}$$

with $\kappa := 4\alpha \left\lceil \log \frac{\Phi_0}{\alpha C_\phi} \right\rceil + \log \frac{\Phi_0}{\eta}$.

Proof. From the observations above, it is clear that the duality gap at the output point $\bar{u}$ is bounded by η.

Also observe that the procedure calls LOsep_X once per iteration. In order to demonstrate the bound in (19), we split the LCG procedure into two phases, and bound the number of iterations separately for each phase. Let C_ϕ denote the curvature of the smooth convex function ϕ.

We say Procedure 1 is in the first phase whenever $\Phi_{t-1} > \eta$. In view of Theorem 5.1 in (Braun et al., 2016), it is clear that the number of iterations in the first phase can be bounded as

$$T_1 \leq 4\alpha \left\lceil \log \frac{\Phi_0}{\alpha C_\phi} \right\rceil + \frac{4\alpha^2 C_\phi}{\eta} + \log \frac{\Phi_0}{\eta}.$$

Procedure 1 enters the second phase when $\Phi_{t-1} \leq \eta$. Again with the argumentation in Theorem 5.1 in (Braun et al., 2016), we obtain that the total number of positive calls in this phase can be bounded by $\frac{4\alpha^2 C_\phi}{\eta}$, if $\eta < \alpha C_\phi$, or by 4α if $\eta \geq \alpha C_\phi$. Moreover, the procedure exits whenever the current LOsep_X oracle call is a negative call. Hence, the number of iterations in the second phase can be bounded by

$$T_2 \leq \begin{cases} \frac{4\alpha^2 C_\phi}{\eta} + 1, & \eta < \alpha C_\phi; \\ 4\alpha + 1, & \eta \geq \alpha C_\phi. \end{cases}$$

Thus, our bound in (19) can be obtained from the above two bounds plus one more LO oracle call at Line 1. □

2.3. The Convergence Properties of CALSGD

This subsection is devoted to establishing the main convergence properties of the CALSGD method. Since the algorithm is stochastic, we will establish the convergence results for finding a stochastic ϵ-solution, i.e., a point $\bar{x} \in X$ s.t. $\mathbb{E}[f(\bar{x}) - f(x^*)] \leq \epsilon$. We first state a simple technical result from (Lan and Zhou, 2014, Lemma 2.1) that we will use.

Lemma 2.2. *Let $w_t \in (0, 1]$, $t = 1, 2, \ldots$, be given. Also let us denote*

$$W_t := \begin{cases} 1 & t = 1 \\ (1 - w_t) W_{t-1} & t \geq 2. \end{cases}$$

Suppose that $W_t > 0$ for all $t \geq 2$ and that the sequence $\{\delta_t\}_{t \geq 0}$ satisfies

$$\delta_t \leq (1 - w_t)\delta_{t-1} + B_t, \quad t = 1, 2, \ldots.$$

Then for any $1 \leq l \leq k$, we have

$$\delta_k \leq W_k \left(\tfrac{1 - w_l}{W_l} \delta_{l-1} + \textstyle\sum_{i=l}^k \tfrac{B_i}{W_i} \right).$$

Theorem 2.3 describes the main convergence properties of the CALSGD method (cf. Algorithm 1). The proof of this theorem can be found in the Supplementary material A.

Theorem 2.3. *Let Γ_k be defined as follows,*

$$\Gamma_k := \begin{cases} 1 & k = 1 \\ \Gamma_{k-1}(1 - \gamma_k) & k \geq 2. \end{cases} \tag{20}$$

Suppose that $\{\beta_k\}$ and $\{\gamma_k\}$ in the CALSGD algorithm satisfy

$$\gamma_1 = 1 \ \text{ and } \ L\gamma_k \leq \beta_k, \ \ k \geq 1. \tag{21}$$

a) If

$$\tfrac{\beta_k \gamma_k}{\Gamma_k} \geq \tfrac{\beta_{k-1}\gamma_{k-1}}{\Gamma_{k-1}}, \ \ k \geq 2, \tag{22}$$

then under assumptions (9) and (10), we have

$$\mathbb{E}\left[f(y_k) - f(x^*)\right] \leq \tfrac{\beta_k \gamma_k}{2} D_X^2 + \Gamma_k \sum_{i=1}^k \left[\tfrac{\eta_i \gamma_i}{\Gamma_i} + \tfrac{\gamma_i \sigma^2}{2\Gamma_i B_i (\beta_i - L\gamma_i)} \right], \tag{23}$$

where x^* is an arbitrary optimal solution of (8) and D_X is defined in (4).

b) If

$$\tfrac{\beta_k \gamma_k}{\Gamma_k} \leq \tfrac{\beta_{k-1}\gamma_{k-1}}{\Gamma_{k-1}}, \ \ k \geq 2, \tag{24}$$

(rather than (36)) is satisfied, then the result in part a) holds by replacing $\beta_k \gamma_k D_X^2$ with $\beta_1 \Gamma_k \|x_0 - x^*\|^2$ in the first term of the RHS of (37).

c) Under the assumptions in part a) or b), the number of inner iterations performed at the k-th outer iterations is bounded by

$$T_k = \begin{cases} \kappa + \frac{8\alpha^2\beta_k D_X^2}{\eta_k} + 2, & \eta_k < \alpha\beta_k D_X^2; \\ \kappa + 4\alpha + \frac{4\alpha^2\beta_k D_X^2}{\eta_k} + 2, & \eta_k \geq \alpha\beta_k D_X^2, \end{cases} \tag{25}$$

with $\kappa := 4\alpha \left\lceil \log \frac{\Phi_0^k}{\alpha\beta_k D_X^2} \right\rceil + \log \frac{\Phi_0^k}{\eta_k}$.

Now we provide two different sets of parameters $\{\beta_k\}, \{\gamma_k\}, \{\eta_k\}$, and $\{B_k\}$, which lead to optimal complexity bounds on the number of calls to the SFO and LOsep_X oracles.

Corollary 2.4. *Suppose that $\{\beta_k\}, \{\gamma_k\}$, $\{\eta_k\}$, and $\{B_k\}$ in the CALSGD method are set to*

$$\beta_k = \tfrac{4L}{k+2}, \ \gamma_k = \tfrac{3}{k+2}, \ \eta_k = \tfrac{LD_X^2}{k(k+1)}, \\ \text{and } B_k = \left\lceil \tfrac{\sigma^2(k+2)^3}{L^2 D_X^2} \right\rceil, \ k \geq 1, \tag{26}$$

and we assume $\|f'(x^*)\|$ is bounded for any optimal solution x^* of (8). Under assumptions (9) and (10), we have

$$\mathbb{E}\left[f(y_k) - f(x^*)\right] \leq \tfrac{6LD_X^2}{(k+2)^2} + \tfrac{9LD_X^2}{2(k+1)(k+2)}, \ \forall k \geq 1. \tag{27}$$

As a consequence, the total number of calls to the SFO *and* LOsep_X *oracles performed by the CALSGD method for finding a stochastic ϵ-solution of (1), respectively, can be bounded by*

$$\mathcal{O}\left\{\sqrt{\tfrac{LD_X^2}{\epsilon}} + \tfrac{\sigma^2 D_X^2}{\epsilon^2}\right\}, \tag{28}$$

and

$$\mathcal{O}\left\{\sqrt{\tfrac{LD_X^2}{\epsilon}} \log \tfrac{LD_X^2}{\Lambda\epsilon} + \tfrac{LD_X^2}{\epsilon}\right\} \text{ with probability } 1-\Lambda. \tag{29}$$

Proof. It can be easily seen from (26) that (35) holds. Also note that by (26), we have

$$\Gamma_k = \tfrac{6}{k(k+1)(k+2)}, \tag{30}$$

and hence

$$\tfrac{\beta_k\gamma_k}{\Gamma_k} = \tfrac{2Lk(k+1)}{k+2},$$

which implies that (36) holds. It can also be easily checked from (30) and (26) that

$$\textstyle\sum_{i=1}^k \tfrac{\eta_i\gamma_i}{\Gamma_i} \leq \tfrac{kLD_X^2}{2}, \ \sum_{i=1}^k \tfrac{\gamma_i}{\Gamma_i B_i(\beta_i - L\gamma_i)} \leq \tfrac{kLD_X^2}{2\sigma^2}.$$

Using the bound in (37), we obtain (27), which implies that the total number of outer iterations N can be bounded by $\mathcal{O}\left(\sqrt{LD_X^2/\epsilon}\right)$ under the assumptions (9) and (10). The bound in (28) then immediately follows from this observation and the fact that the number of calls to the SFO oracle is bounded by

$$\textstyle\sum_{k=1}^N B_k \leq \sum_{k=1}^N \frac{\sigma^2(k+2)^3}{L^2 D_X^2} + N \leq \frac{\sigma^2(N+3)^4}{4L^2 D_X^2} + N.$$

We now provide a good estimation for Φ_0^k (cf. Line 1 in LCG procedure) at the k-th outer iteration. In view of the definition of Φ_0^k and $\psi(\cdot)$ (cf. (16)), we have,

$$\Phi_0^k = \langle \psi_k'(x_{k-1}), x_{k-1} - x\rangle = \langle g_k, x_{k-1} - x\rangle.$$

Moreover, let $A_k := \|g_k - f'(z_k)\|_* \geq \sqrt{\frac{N\sigma^2}{\Lambda B_k}}$, by Chebyshev's inequality and (15), we obtain,

$$\text{Prob}\{A_k\} \leq \tfrac{\mathbb{E}[\|g_k - f'(z_k)\|_*^2]\Lambda B_k}{N\sigma^2} \leq \tfrac{\Lambda}{N}, \ \forall \Lambda < 1, k \geq 1,$$

which implies that $\text{Prob}\{\bigcap_{k=1}^N \bar{A}_k\} \leq 1 - \Lambda$. Hence, by Cauchy-Schwarz and triangle inequalities, we have with probability $1 - \Lambda$,

$$\begin{aligned} \Phi_0^k &= \langle g_k - f'(z_k), x_{k-1} - x\rangle + \langle f'(z_k), x_{k-1} - x\rangle\} \\ &\leq \left(\sqrt{\tfrac{N\sigma^2}{\Lambda B_k}} + \|f'(z_k) - f'(x^*)\|_* + \|f'(x^*)\|_*\right) D_X \\ &\leq \left(\sqrt{\tfrac{N}{\Lambda k^3}} + 1\right) LD_X^2 + \|f'(x^*)\|_* D_X, \end{aligned} \tag{31}$$

where the last inequality follows from (6) and (26).

Note that we always have $\eta_k < \alpha\beta_k D_X^2$. Therefore, it follows from the bound in (39), (26), and (31) that the total number of inner iterations can be bounded by

$$\begin{aligned} \textstyle\sum_{k=1}^N T_k &\leq \textstyle\sum_{k=1}^N \left[4\alpha\left(\log \frac{\Phi_0^k}{\alpha\beta_k D_X^2} + 1\right) + \log \frac{\Phi_0^k}{\eta_k} \right. \\ &\quad \left. + \tfrac{8\alpha^2\beta_k D_X^2}{\eta_k} + 2\right] \\ &\leq \sum_{k=1}^N \left[5\alpha \log\left(2k^2\left(\sqrt{\tfrac{N}{\Lambda k^3}} + 1 + \tfrac{\|f'(x^*)\|_*}{LD_X}\right)\right)\right. \\ &\quad \left. + 32\alpha^2 k\right] + (4\alpha + 2)N \\ &= \mathcal{O}(N \log \tfrac{N^2}{\Lambda} + N^2 + N), \end{aligned}$$

which implies that our bound in (29). □

We now provide a slightly improved complexity bound on the number of calls to the SFO oracle which depends on the distance from the initial point to the set of optimal solutions, rather than the diameter D_X. In order to obtain this improvement, we need to estimate $D_0 \geq \|x_0 - x^*\|$ and to fix the number of iterations N in advance. This result will play an important role for the analysis of the CALSGD method to solve strongly convex problems (see Supplementary Material C.1).

Corollary 2.5. *Suppose that there exists an estimate D_0 s.t. $\|x_0 - x^*\| \leq D_0 \leq D_X$. Also assume that the outer iteration limit $N \geq 1$ is given. If*

$$\beta_k = \tfrac{3L}{k}, \ \gamma_k = \tfrac{2}{k+1}, \ \eta_k = \tfrac{2LD_0^2}{Nk}, \quad \text{and} \ B_k = \left\lceil \tfrac{\sigma^2 N(k+1)^2}{L^2 D_0^2} \right\rceil, \ k \geq 1. \tag{32}$$

Under assumptions (9) and (10),

$$\mathbb{E}\left[f(y_N) - f(x^*)\right] \leq \tfrac{8LD_0^2}{N(N+1)}, \ \forall N \geq 1.$$

As a consequence, the total number of calls to the SFO *and* LOsep_X *oracles performed by the CALSGD method for finding a stochastic ϵ-solution of (1), respectively, can be bounded by*

$$\mathcal{O}\left\{\sqrt{\tfrac{LD_0^2}{\epsilon}} + \tfrac{\sigma^2 D_0^2}{\epsilon^2}\right\}, \tag{33}$$

and (29).

Proof. The proof is similar to Corollary 2.4, and hence details are skipped. □

It should be pointed out that the complexity bound for the number of calls to the LOsep oracle in (29) is established with probability $1 - \Lambda$. However, the probability parameter Λ only appears in the non-dominant term.

3. Experimental Results

We present preliminary experimental results showing the performance of CALSGD compared to OFW for stochastic optimization. As examples we use the video co-localization problem, which can be solved by quadratic programming over a path polytope, different structured regression problems, and quadratic programming over the standard spectrahedron. In all cases we use objective functions of the form $\|Ax - b\|^2$, with $A \in \mathbb{R}^{m \times n}$, i.e., m examples over a feasible region of dimension n. For comparability we use a batch size of 128 for all algorithms to compute each gradient and the full matrix A for the actual objective function values. All graphs show the function value using a logscale on the vertical axis.

In Figure 1 we compare the performance of three algorithms: CALSGD, SCGS and OFW. As described above SCGS is the non-lazy counterpart of CALSGD. In the four graphs of Figure 1 we report the objective function value over the number of iterations, the wall clock time in seconds, the number of calls to the linear oracle, and the number of gradient evaluations in that order. In all these measures, our proposed algorithms outperform OFW by multiple orders of magnitude. As expected in number of iterations and number of gradient evaluations both versions CALSGD and SCGS perform equally well, however in wall clock time and in the number of calls to the linear oracle we observe the advantage of the weaker LOsep oracle over LO.

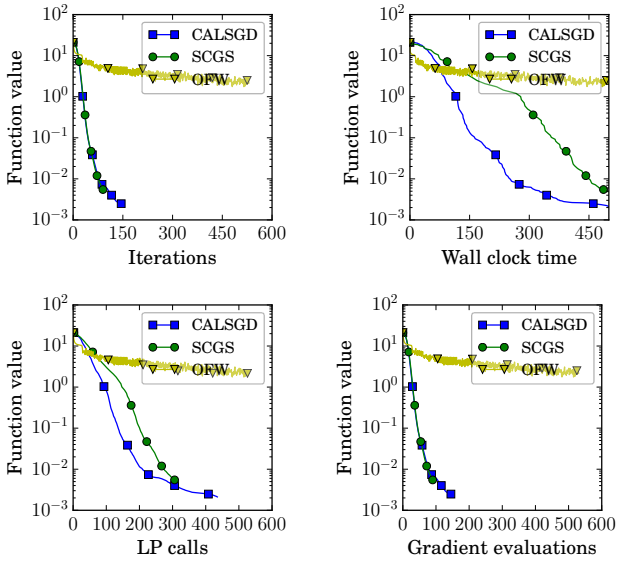

Figure 1. Performance of CALSGD and its non-lazy variant SCGS on a structured regression problem compared to OFW. The feasible region is the flow-based formulation of the convex hull of Hamiltonian cycles on 9 nodes and has dimension $n = 162$.

In Figure 5 and 4 we show the performance of CALSGD on one video co-localization instance and one semi-definite convex programming instance. Due to space limitations we only report the function value over the number of iterations and wall clock time in seconds; see Supplementary Material D for a detailed analysis as well as more examples.

Implementation details. Finally, we provide details of the implementation of LOsep. In the case of the structured regression problems and the quadratic optimizations over the path polytope instances, we used Gurobi as a solver and included callbacks to terminate whenever the required improvement (given by Φ_{t-1}) is reached; our approach is one out of many and other approaches could have been used equally well. If the solver does not find a good enough solution, it returns with a lower bound on the Wolfe gap, which we use to update Φ_t. In the case of convex programming over the feasible region $S_n := \{X \in R^{n \times n} \mid X \succcurlyeq 0 \text{ and } tr(X) = 1\}$, we compute a maximal eigenvector of the gradient (which is a matrix in this case) and use the rank-1 factor of the maximal eigenvector, which is an optimal point. In this case, there is no early termination.

However, in all cases, we use caching, i.e., we store all previously seen points and check if any of them satisfies the improvement guarantee. If that is the case we do not call Gurobi or the maximal eigenvector routine. The size of the cache is very small in all experiments; alternatively one could use cache strategies such as e.g., k-paging.

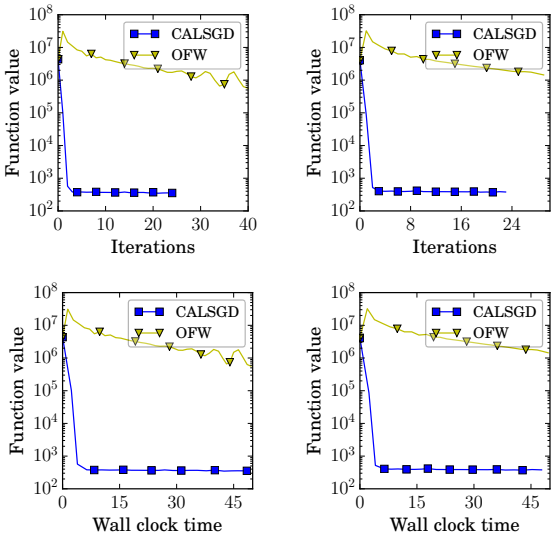

Figure 2. Two small video co-localization instances. On the left: `road_paths_01_DC_a` instance ($n = 29682$ and $m = 10000$). On the right: `road_paths_01_DC_b` instance ($n = 29682$ and $m = 10000$). Observe a significant difference in function value of multiple orders of magnitude after a few seconds.

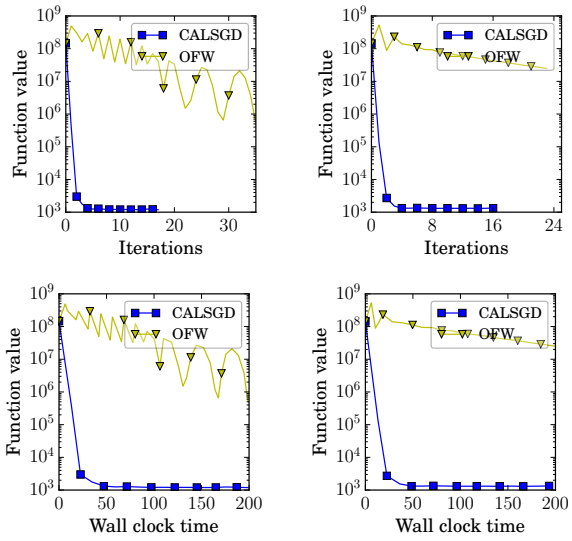

Figure 3. Two medium sized video co-localization instances. On the left: `road_paths_02_DE_a` instance ($n = 119520$ and $m = 10000$). On the right: `road_paths_02_DE_b` instance ($n = 119520$ and $m = 10000$). Similar results as in Figure 2.

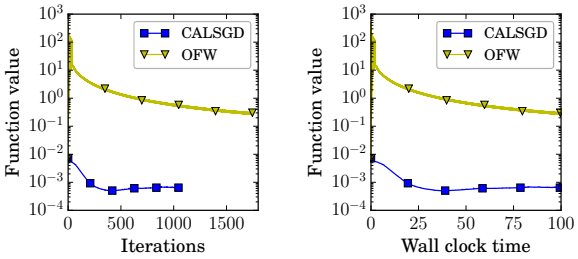

Figure 4. Performance of CALSGD and OFW on a medium sized convex programming instance with feasible region $S_n := \{X \in \mathbb{R}^{n \times n} \mid X \succcurlyeq 0, tr(X) = 1\}$ with $n = 100$ and $m = 10000$. Similar to the results before in both iterations and wall clock time our method performs better.

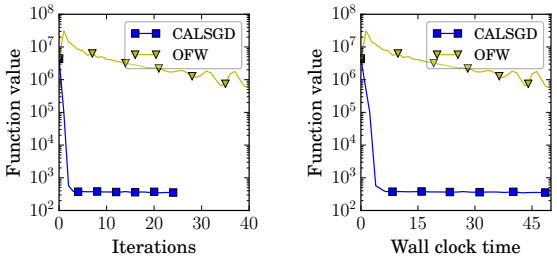

Figure 5. Performance of CALSGD compared to OFW on a small video co-localization instance. The dimension of the underlying path polytope is $n = 29682$, the time limit is 50 seconds. Our algorithm performs significantly better both in number of iterations as well as in wall clock time.

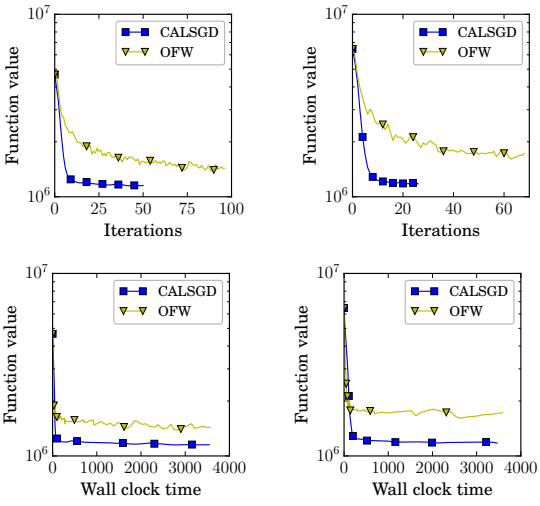

Figure 6. Structured regression problem over the convex hull of all Hamiltonian cycles of a graph on 11 nodes ($n = 242$) on the left and 12 nodes ($n = 288$) on the right. We used a density of $d = 0.6$ for A and $m = 10000$. On both instances CALSGD achieves lower values much faster, both in number of iterations as well as in wall clock time.

Acknowledgements

We would to thank Elad Hazan for providing references. Research reported in this paper was partially supported by NSF CAREER award CMMI-1452463.

References

S. Ahipasaoglu and M. Todd. A Modified Frank-Wolfe Algorithm for Computing Minimum-Area Enclosing Ellipsoidal Cylinders: Theory and Algorithms. *Computational Geometry*, 46:494–519, 2013.

F. Bach, S. Lacoste-Julien, and G. Obozinski. On the equivalence between herding and conditional gradient algorithms. In *the 29th International Conference on Machine Learning*, 2012.

A. Beck and M. Teboulle. A conditional gradient method with linear rate of convergence for solving convex linear systems. *Math. Methods Oper. Res.*, 59:235–247, 2004.

G. Braun, S. Pokutta, and D. Zink. Lazifying conditional gradient algorithms. *arXiv preprint arXiv:1610.05120*, 2016.

Y. Chen, G. Lan, and Y. Ouyang. Optimal primal-dual methods for a class of saddle point problems. *SIAM Journal on Optimization*, 24(4):1779–1814, 2014.

K. L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Trans. Algorithms*, 6(4): 63:1–63:30, Sept. 2010.

B. Cox, A. Juditsky, and A. S. Nemirovski. Dual subgradient algorithms for large-scale nonsmooth learning problems. Manuscript, School of ISyE, Georgia Tech, Atlanta, GA, 30332, USA, 2013. submitted to *Mathematical Programming*, Series B.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

R. M. Freund and P. Grigas. New Analysis and Results for the Frank-Wolfe Method. *ArXiv e-prints*, July 2013.

D. Garber and E. Hazan. A Linearly Convergent Conditional Gradient Algorithm with Applications to Online and Stochastic Optimization. *ArXiv e-prints*, Jan 2013.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, I: a generic algorithmic framework. *SIAM Journal on Optimization*, 22:1469–1492, 2012.

S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23:2061–2089, 2013.

Gurobi Optimization. Gurobi optimizer reference manual version 6.5, 2016. URL https://www.gurobi.com/documentation/6.5/refman/.

Z. Harchaoui, A. Juditsky, and A. S. Nemirovski. Conditional gradient algorithms for machine learning. NIPS OPT workshop, 2012.

E. Hazan. Sparse approximate solutions to semidefinite programs. In E. Laber, C. Bornstein, L. Nogueira, and L. Faria, editors, *LATIN 2008: Theoretical Informatics*, volume 4957 of *Lecture Notes in Computer Science*, pages 306–316. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-78772-3.

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

E. Hazan and H. Luo. Variance-reduced and projection-free stochastic optimization. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1263–1271, 2016.

M. Jaggi. *Sparse Convex Optimization Methods for Machine Learning*. PhD thesis, ETH Zürich, 2011. http://dx.doi.org/10.3929/ethz-a-007050453.

M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *the 30th International Conference on Machine Learning*, 2013.

M. Jaggi and M. Sulovský. A simple algorithm for nuclear norm regularized problems. In *the 27th International Conference on Machine Learning*, 2010.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *European Conference on Computer Vision*, pages 253–268. Springer, 2014.

G. Lan. Convex optimization under inexact first-order information. Ph.D. dissertation, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, 2009.

G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1):365–397, 2012.

G. Lan. The complexity of large-scale convex programming under a linear optimization oracle. *Technical Report*, 2013. Available on http://www.optimization-online.org/.

G. Lan and Y. Zhou. Conditional gradient sliding for convex optimization. *Optimization-Online preprint (4605)*, 2014.

R. Luss and M. Teboulle. Conditional gradient algorithms for rank one matrix approximations with a sparsity constraint. *SIAM Review*, 55:65–98, 2013.

M. Mahdavi, L. Zhang, and R. Jin. Mixed optimization for smooth functions. In *Advances in Neural Information Processing Systems*, pages 674–682, 2013.

Y. E. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Doklady AN SSSR*, 269:543–547, 1983.

Y. E. Nesterov. *Introductory Lectures on Convex Optimization: a basic course*. Kluwer Academic Publishers, Massachusetts, 2004.

Y. E. Nesterov. Smooth minimization of nonsmooth functions. *Mathematical Programming*, 103:127–152, 2005.

C. Shen, J. Kim, L. Wang, and A. van den Hengel. Positive semidefinite metric learning using boosting-like algorithms. *Journal of Machine Learning Research*, 13: 1007–1036, 2012.