Conditional Gradient Methods via Stochastic Path-Integrated Differential Estimator

Alp Yurtsever [1] **Suvrit Sra** [2] **Volkan Cevher** [1]

Abstract

We propose a class of novel variance-reduced stochastic conditional gradient methods. By adopting the recent stochastic path-integrated differential estimator technique (SPIDER) of Fang et al. (2018) for the classical Frank-Wolfe (FW) method, we introduce SPIDER-FW for finite-sum minimization as well as the more general expectation minimization problems. SPIDER-FW enjoys superior complexity guarantees in the non-convex setting, while matching the best known FW variants in the convex case. We also extend our framework à la conditional gradient sliding (CGS) of Lan & Zhou (2016), and propose SPIDER-CGS.

1. Introduction

We study two different problem settings in this paper, *finite-sum* and the more general *expectation minimization*:

$$\underset{x\in\Omega}{\text{minimize}} \quad F(x) := \begin{cases} \mathbb{E}_\xi f(x,\xi) & \text{(expectation)} \\ \frac{1}{n}\sum_{i=1}^n f_i(x) & \text{(finite-sum)} \end{cases} \tag{1}$$

▹ $\Omega \subset \mathbb{R}^d$ is the convex and compact domain;
▹ F, f and f_i are differentiable and *possibly non-convex*;
▹ $\xi \sim \mathcal{P}$ is a random variable, supported on $\Xi \subset \mathbb{R}^p$.

The expectation objective template covers a large number of applications in machine learning and statistics. The finite-sum template frequently arises in M-estimation and empirical risk minimization problems. Accordingly, there are many applications for stochastic conditional gradient methods both in convex and non-convex settings. This includes low-rank matrix and tensor factorizations, structured sparse matrix estimation, dictionary learning applications, multi-class classification (considered as a motivating example in a related work by Hazan & Luo (2016)), constrained deep learning problems (*e.g.,* Ravi et al. (2018) present an application in computer vision) and many more.

Template (1) can be solved by using the well-known projected stochastic gradient descent method (SGD). At each iteration, SGD takes a stochastic gradient step followed by a projection to ensure the feasibility of the new point. However, in many applications, projection onto Ω can impose a computational bottleneck (*e.g.,* projection onto the nuclear norm-ball may require a full singular value decomposition), or it can be even intractable (*e.g.,* dual structural SVMs (Lacoste-Julien et al., 2013)).

As a result, the Frank-Wolfe (FW) algorithm (*aka* conditional gradient method) has witnessed tremendous interest in the machine learning community in the last decade. FW avoids projection by leveraging the so-called linear minimization oracle instead:

$$\mathrm{lmo}_\Omega(v) = \arg\min_{x\in\Omega} \langle x, v\rangle. \tag{\textit{lmo}}$$

lmo is significantly cheaper to compute than projection. For instance, *lmo* of nuclear norm-ball requires the computation of the leading singular vectors only (*vs.* the full spectrum for projection), which can be efficiently found by using Krylov subspace methods (Jaggi, 2013).

Our focus in this paper is on the theoretical complexity of stochastic and finite-sum FW, with an aim to identify and present the tightest results known so far. To this end, we also propose a class of novel variance-reduced stochastic optimization algorithms, based on the recent *stochastic path-integrated differential estimator* technique (SPIDER) of Fang et al. (2018).

By combining SPIDER with the classical FW method, we introduce SPIDER-FW for finite-sum and expectation minimization problems. We also extend our framework à la conditional gradient sliding (CGS) of Lan & Zhou (2016), and propose SPIDER-CGS.

From SPIDER, we adopt the variance bounds from Lemma 1 of (Fang et al., 2018), which relates the variance of the current estimator to the error of the previous estimator and the distance between the iterates. Nevertheless, Fang et al. (2018) introduce SPIDER for normalized gradient method

[1]Ecole Polytechnique Fédérale de Lausanne, Switzerland [2]Massachusetts Institute of Technology, USA. Correspondence to: Alp Yurtsever <alp.yurtsever@epfl.ch>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

which is fundamentally different than the FW method. Accordingly, the analyses are different.

A natural and widely used measure for the convergence of conditional gradient methods is the so-called FW-gap (*cf.,* Section 3). However, we are not aware of any reported FW-gap convergence of CGS in the non-convex settings. Therefore, we present a new compact proof (and an extension for the stochastic setting) in the supplementary material. Although CGS does not seem to provide any improvement upon FW in this setup, we use the proof technique to extend SPIDER-CGS for the non-convex settings.

Finally, for the majority of the variance reduced FW methods in the literature, the analysis relies on the induction technique with respect to the outer loop counter, along with a sufficient improvement condition for each epoch. Consequently, at the beginning of each epoch parameters are typically reset. Instead, we set our learning-rate parameters with respect to the more natural total iteration counter, and we go over the proof without induction.

Roadmap. Section 2 provides an extensive discussion on the related works. Section 3 recalls some basic notions from the optimization theory. Sections 4 and 5 present SPIDER-FW and SPIDER-CGS respectively, along with their theoretical guarantees for various problem settings. Section 6 provides an extensive comparison of the theoretical complexity of FW methods in the literature. Finally, Section 7 draws the conclusions.

Notation. We work on the real space with Euclidean norms for simplicity. Throughout, $\langle \cdot, \cdot \rangle$ represents the standard inner product associated with the Euclidean norm $\| \cdot \|$. We use the notation $[n] = \{1, 2, \ldots, n\}$. D denotes diameter of Ω, *i.e.,* $D = \max_{(x,y)\in\Omega^2} \|x - y\|$.

2. Related Works

Frank-Wolfe algorithm. This classical method is first proposed by Frank & Wolfe (1956) for solving smooth convex minimization problems with a polyhedral domain constraint (polyhedral constraint is relaxed for an arbitrary convex compact set by Jaggi (2013)).

Algorithm 1 Frank-Wolfe algorithm

Input: $x^1 \in \Omega$
for $k = 1, 2, \ldots, K$ **do**
  Compute $w^k \in \mathrm{lmo}_\Omega \left(\nabla F(x^k) \right)$
  Update $x^{k+1} = x^k + \eta_k \left(w^k - x^k \right)$
end for

Given an initial guess $x^1 \in \Omega$, at each iteration, FW minimizes the linear approximation of F at the current iterate x^k over Ω (this corresponds to the *lmo* step). Clearly, minimization of a linear function returns an extreme point of the domain. Since the new estimate is constructed as a convex combination of the current iterate and this extreme point, by definition it is a feasible point, hence the method does not require projections.

FW did not attract much attention in the machine learning community due to its slow convergence rate until Hazan (2008) and Jaggi (2013) emphasize the favorable trade-off between the convergence rate and the per-iteration cost provided by FW in key applications. Following then, there has been a resurgence of interest for FW-type algorithms.

FW literature in the stochastic optimization setting is much younger compared to the projection-based stochastic gradient methods. We can trace it back to a variant for online learning proposed by Hazan & Kale (2012). More recently, Hazan & Luo (2016) introduced stochastic FW methods with and without variance reduction for finite-sum problems. Very recently, Mokhtari et al. (2018) have proposed an alternative scheme for expectation minimization setting.

FW methods for non-convex stochastic learning are relatively understudied, most of the known results are due to Reddi et al. (2016). We discuss more details on the theoretical aspects of all these FW variants in Section 6.

Conditional gradient sliding. Lan & Zhou (2016) has recently developed the conditional gradient sliding method (CGS) based on the idea of applying accelerated gradient method (AG) of Nesterov (1983) for solving problems from template (1), but applying FW to the projection subproblems. In other words, CGS establishes the convergence of an inexact version of AG. Surprisingly, CGS has superior first-order oracle complexity compared to FW, although they have the same *lmo* complexity. We discuss more details and variants of CGS in Section 6.

SPIDER. There has been extensive research on variance reduced stochastic optimization methods in order to address the needs of machine learning and big data applications. Therefore, various variance reduction techniques are proposed in the last few years such as SAG (Roux et al., 2012), SVRG (Johnson & Zhang, 2013), SAGA (Defazio et al., 2014), and more recently SARAH (Nguyen et al., 2017) and SPIDER (Fang et al., 2018).

SARAH and SPIDER are closely related since they use the same sequential update rule for the gradient estimator v^k:

$$v^k = \nabla f_{\mathcal{S}}(x^k) - \nabla f_{\mathcal{S}}(x^{k-1}) + v^{k-1}.$$

However, SARAH uses this estimator in the classical gradient descent template, while SPIDER adopts a normalized gradient approach, and their results and analyses differ.

As described by Wang et al. (2018), the original SPIDER framework has a restrictive step-size (proportional with the target accuracy ϵ), which makes the algorithm impractical though its theoretical appeal. Surprisingly, this problem disappears in the conditional gradient framework analysis.

3. Preliminaries

Solution. We denote a solution and the optimal value of problem (1) by $x^\star$ and $F^\star$ respectively:

$$x^\star \in \arg\min_{x\in\Omega} F(x) \quad \text{and} \quad F^\star = F(x^\star).$$

The measure of non-stationarity. For unconstrained non-convex problems, the typical measure of non-stationarity is the gradient norm, because $\|\nabla f(x)\| \to 0$ as x converges to a stationary point. However, this measure cannot be used for constrained problems, because $\|\nabla f(x)\|$ might not converge to 0 when we approach to a solution on the boundary.

Instead, we will use the quantity

$$\mathcal{G}(x) := \max_{u\in\Omega} \langle u - x, -\nabla F(x)\rangle ,$$

which is widely known as the FW gap (because it naturally appears in the analysis of FW-type methods). FW gap is always non-negative, and it gets 0 if and only if we are looking at a stationary point or a solution. Therefore, FW-gap is a meaningful measure of non-stationarity. It was also used by Lacoste-Julien (2016) and Reddi et al. (2016).

ϵ-solution. Due to the fundamental difference in the measure of non-stationarity, we use different definitions of approximate solutions for convex and non-convex problems:

▷ If F is *convex*, we say $x_\epsilon^\star \in \Omega$ is an ϵ-solution if

$$F(x_\epsilon^\star) - F^\star \le \epsilon.$$

▷ If F is *non-convex*, we say that a random variable $x_\epsilon^\star$ chosen uniformly from a finite set of points $\{x^1, x^2, \ldots, x^k\}$ is an ϵ-solution if

$$\mathbb{E}[\mathcal{G}(x_\epsilon^\star)] \le \epsilon.$$

It is common to provide convergence guarantees in expectation for a randomly chosen iterate in the non-convex setting. See (Reddi et al., 2016).

Oracle models. We adopt the following black-box oracle model from Reddi et al. (2016), to establish a ground for comparing the convergence speed of different algorithms:

- Stochastic first-order oracle (*sfo*)
  For a stochastic function $\mathbb{E}_\xi f(\,\cdot\,,\xi)$ with $\xi \sim \mathcal{P}$, *sfo* returns a pair $(f(x,\xi'), \nabla f(x,\xi'))$ where ξ' is an *iid* sample from $\mathcal{P}$. (Nemirovski & Yudin, 1983)
- Incremental first-order oracle (*ifo*)
  For a finite-sum, *ifo* takes an index $i \in [n]$ and returns $(f_i(x), \nabla f_i(x))$. (Agarwal & Bottou, 2014)
- Linear minimization oracle (*lmo*)
  Well-known oracle of FW-type methods.

Assumptions (finite-sum). For the finite-sum setting, we assume that $f_i(x)$ has an averaged L-Lipschitz gradient:

$$\mathbb{E}\big\|\nabla f_i(x) - \nabla f_i(y)\big\|^2 \le L^2\big\|x-y\big\|^2, \;\; \forall (x,y)\in\Omega^2.$$

Note that this implies F is L-smooth, since

$$\begin{aligned}\big\|\nabla F(x) - \nabla F(y)\big\|^2 &= \big\|\mathbb{E}(\nabla f_i(x) - \nabla f_i(y))\big\|^2 \\ &\le \mathbb{E}\big\|\nabla f_i(x) - \nabla f_i(y)\big\|^2 \le L^2\big\|x-y\big\|^2.\end{aligned}$$

Assumptions (expectation). For the expectation minimization, we assume that $\nabla f(x,\xi)$ is an unbiased estimate of the gradient:

$$\mathbb{E}\nabla f(x,\xi) = \nabla F(x).$$

We also assume that the variance is bounded:

$$\mathbb{E}\big\|\nabla f(x,\xi) - \nabla F(x)\big\|^2 \le \sigma^2 < \infty, \quad \forall \xi \in \Xi,\; \forall x \in \Omega.$$

And finally, we assume an averaged L-Lipschitz gradient condition, *i.e.,* the following condition holds $\forall \xi \in \Xi$:

$$\mathbb{E}\big\|\nabla f(x,\xi) - \nabla f(y,\xi)\big\|^2 \le L^2\big\|x-y\big\|^2, \;\; \forall (x,y)\in\Omega^2.$$

Similar to the finite-sum, this implies the smoothness of F.

Assumptions (non-convex). Let us denote the initial point by $\bar{x}^1$. Initial suboptimality $F(\bar{x}^1) - F^\star$ appears in the convergence bounds in the non-convex setting. For notational convenience, we denote an upper bound on this term by $\mathcal{E}$:

$$F(\bar{x}^1) - F^\star \le \mathcal{E}$$

Assume that $F^\star$ is finite, then there exists a finite $\mathcal{E}$ which satisfies this bound. This is a direct consequence of the smoothness of F and the boundedness of domain.

All these assumptions are mild and frequently used in the analysis of stochastic methods and FW-type algorithms.

4. SPIDER Frank-Wolfe

This section presents SPIDER-FW algorithm and its convergence guarantees for various problem settings.

Our methods have a double loop structure, hence the iterates and the parameters have two different iteration counters t and k, such as $x^{t,k}$. For notational simplicity, we drop the first counter when there is no ambiguity, such as x^k. Throughout, $s_{t,k}$ denotes the total number of inner iterations until k^{th} iteration of t^{th} epoch. In our pseudocodes, *draw samples* means *iid* samples for expectation minimization, and uniform selection with replacement in the finite-sum.

Algorithm 2 SPIDER Frank-Wolfe

Input: $\bar{x}^1 \in \Omega$
for $t = 1, 2, \ldots, T$ **do**
  Set $x^1 = \bar{x}^t$
  Draw Q_t samples $\mathcal{Q}_t$
  Compute $v^1 = \nabla f_{\mathcal{Q}_t}(x^1)$
  Compute $w^1 \in \mathrm{lmo}_\Omega(v^1)$
  Update $x^2 = x^1 + \eta_{t,1}(w^1 - x^1)$
  for $k = 2, 3, \ldots, K_t$ **do**
    Draw $S_{t,k}$ samples $\mathcal{S}_{t,k}$
    Compute $v^k = \nabla f_{\mathcal{S}_{t,k}}(x^k) - \nabla f_{\mathcal{S}_{t,k}}(x^{k-1}) + v^{k-1}$
    Compute $w^k \in \mathrm{lmo}_\Omega(v^k)$
    Update $x^{k+1} = x^k + \eta_{t,k}(w^k - x^k)$
  end for
  Set $\bar{x}^{t+1} = x^{K_t+1}$
end for

SPIDER-FW: Convex finite-sum

We consider SPIDER-FW with

$$K_t = 2^{t-1} \text{ for } t = 1, 2, \ldots, T.$$

We choose the sampling parameters

$$S_{t,k} = K_t \qquad \mathcal{Q}_t = [n]$$

and the learning rate parameter

$$\eta_{t,k} = \frac{2}{s_{t,k}+1} \text{ where } s_{t,k} = K_t + k - 1.$$

Theorem 1. *Consider the convex finite-sum optimization template, and suppose that the assumptions in Section 3 for this template hold. Then, estimate $x^{t,k}$ of SPIDER-FW with the parameter choices described above satisfies*

$$\mathbb{E}[F(x^{t,k})] - F^\star = \mathcal{O}\left(\frac{LD^2}{s_{t,k}}\right)$$

Corollary 1. *The ifo and lmo complexities of SPIDER-FW for achieving ϵ-solution in this setting are as follows:*

$$\#(\textit{ifo}) \quad = \mathcal{O}\big(n \ln(\tfrac{LD^2}{\epsilon}) + \tfrac{L^2D^4}{\epsilon^2}\big)$$
$$\#(\textit{lmo}) \quad = \mathcal{O}(\tfrac{LD^2}{\epsilon})$$

SPIDER-FW: Convex expectation minimization

We consider SPIDER-FW with

$$K_t = 2^{t-1} \text{ for } t = 1, 2, \ldots, T.$$

We choose the sampling parameters

$$S_{t,k} = K_t \qquad Q_t = \lceil \frac{\sigma^2 K_t^2}{5L^2D^2} \rceil$$

and the learning rate parameter

$$\eta_{t,k} = \frac{2}{s_{t,k}+1} \text{ where } s_{t,k} = K_t + k - 1.$$

Theorem 2. *Consider the convex expectation minimization template, and suppose that the assumptions in Section 3 for this template hold. Then, estimate $x^{t,k}$ of SPIDER-FW with the parameter choices described above satisfies*

$$\mathbb{E}[F(x^{t,k})] - F^\star = \mathcal{O}\left(\frac{LD^2}{s_{t,k}}\right)$$

Corollary 2. *The sfo and lmo complexities of SPIDER-FW for achieving ϵ-solution in this setting are as follows:*

$$\#(\textit{sfo}) \quad = \mathcal{O}\big(\tfrac{\sigma^2D^2 + L^2D^4}{\epsilon^2}\big)$$
$$\#(\textit{lmo}) \quad = \mathcal{O}(\tfrac{LD^2}{\epsilon})$$

SPIDER-FW has the same asymptotic oracle complexities as SCGS (Lan & Zhou, 2016) in this setting. In Section 5, we also present the SPIDER-CGS.

SPIDER-FW: Non-convex finite-sum

We consider SPIDER-FW with

$$K_t = K = \lceil \sqrt{n} \rceil.$$

Furthermore, we choose the parameters as

$$S_{t,k} = S = \lceil \sqrt{n} \rceil \qquad \mathcal{Q}_t = [n]$$

and the learning rate parameter

$$\eta_{t,k} = \eta = \frac{1}{\sqrt{s_{T,K}}} \text{ where } s_{T,K} = TK.$$

Theorem 3. *Consider the non-convex finite-sum template, and suppose that the assumptions in Section 3 for this template hold. Denote by x^{out} an iterate $x^{t,k}$ of SPIDER-FW chosen uniformly random over all (t,k) pairs up to (T,K). Then, the following bound on the FW-gap holds:*

$$\mathbb{E}[\mathcal{G}(x^{out})] = \mathcal{O}\left(\frac{\mathcal{E} + LD^2}{\sqrt{s_{T,K}}}\right)$$

Although it is impractical to store all estimates until the final iteration, all stochastic methods for the non-convex setting shown in Table 1 have this type convergence guarantees, see (Reddi et al., 2016). More stringently, Lacoste-Julien (2016) proves convergence of non-convex FW in terms of the running best iterate. However, we cannot keep track of best estimate in the stochastic setting, simply because we cannot measure the FW-gap.

Corollary 3. *The ifo and lmo complexities of SPIDER-FW for achieving ϵ-solution in the non-convex finite-sum setting are as follows:*

$$\begin{aligned}\#(\textit{ifo}) &= \mathcal{O}\left(\tfrac{\sqrt{n}}{\epsilon^2}(\mathcal{E}^2+L^2D^4)\right)\\ \#(\textit{lmo}) &= \mathcal{O}\left(\tfrac{1}{\epsilon^2}(\mathcal{E}^2+L^2D^4)\right)\end{aligned}$$

SPIDER-FW has better *ifo* complexity than state-of-the-art in the non-convex finite-sum setting. It improves the dependence on sample size n. See Table 1 for comparison.

SPIDER-FW: Non-convex expectation minimization

We consider SPIDER-FW with

$$K_t = K = \lceil \sigma/\epsilon \rceil.$$

Furthermore, we choose the parameters as

$$S_{t,k} = S = \lceil \sigma/\epsilon \rceil \qquad Q_t = Q = \lceil 4(\sigma/\epsilon)^2 \rceil$$

and the learning rate parameter

$$\eta_{t,k} = \eta = \frac{1}{\sqrt{s_{T,K}}} \quad \text{where} \quad s_{T,K} = TK.$$

Theorem 4. *Consider the non-convex expectation minimization template, and suppose that the assumptions in Section 3 for this template hold. Denote by x^{out} an iterate $x^{t,k}$ of SPIDER-FW chosen uniformly random over all (t, k) pairs up to (T, K). Then, the following bound holds:*

$$\mathbb{E}[\mathcal{G}(x^{out})] = \mathcal{O}\left(\frac{\mathcal{E} + LD^2}{\sqrt{s_{T,K}}}\right) + \frac{\epsilon}{2}$$

Corollary 4. *The sfo and lmo complexities of SPIDER-FW for achieving ϵ-solution in this setting are as follows:*

$$\begin{aligned}\#(\textit{sfo}) &= \mathcal{O}\left(\tfrac{\sigma}{\epsilon^3}(\mathcal{E}^2+L^2D^4)\right)\\ \#(\textit{lmo}) &= \mathcal{O}\left(\tfrac{1}{\epsilon^2}(\mathcal{E}^2+L^2D^4)\right)\end{aligned}$$

Once again, SPIDER-FW enjoys superior *sfo* complexity while maintaining the same *lmo* complexity as its competitors. SVRF was the state-of-the-art with $\mathcal{O}(\epsilon^{-10/3})$, see Reddi et al. (2016).

5. SPIDER Conditional Gradient Sliding

This section presents SPIDER-CGS (as shown in Algorithm 3) and its convergence guarantees for various settings. SPIDER-CGS has the same oracle complexity as the SPIDER-FW.

Algorithm 3 SPIDER Conditional Gradient Sliding

Input: $\bar{x}^1 = \bar{y}^1 \in \Omega$
for $t = 1, 2, \ldots, T$ **do**
  Set $x^1 = \bar{x}^t$ and $y^1 = \bar{y}^t$
  Update $z^1 = y^1 + \gamma_{t,1}(x^1 - y^1)$
  Draw Q_t samples $\mathcal{Q}_t$
  Compute $v^1 = \nabla f_{\mathcal{Q}_t}(z^1)$
  $x^2 = \mathrm{CndG}(x^1, v^1, \alpha_{t,1}, \beta_{t,1})$
  Update $y^2 = y^1 + \gamma_{t,1}(x^2 - y^1)$
  for $k = 2, 3, \ldots, K_t$ **do**
    Update $z^k = y^k + \gamma_{t,k}(x^k - y^k)$
    Draw $S_{t,k}$ samples $\mathcal{S}_{t,k}$
    Compute $v^k = \nabla f_{\mathcal{S}_{t,k}}(z^k) - \nabla f_{\mathcal{S}_{t,k}}(z^{k-1}) + v^{k-1}$
    $x^{k+1} = \mathrm{CndG}(x^k, v^k, \alpha_{t,k}, \beta_{t,k})$
    Update $y^{k+1} = y^k + \gamma_{t,k}(x^{k+1} - y^k)$
  end for
  Set $\bar{x}^{t+1} = x^{K_t+1}$ and $\bar{y}^{t+1} = y^{K_t+1}$
end for

function $u^+ = \mathrm{CndG}(u, v, \alpha, \beta)$
  Set $u^1 = u$
  for $k = 1, 2, \ldots$ **do**
    Compute $w^k \in \mathrm{lmo}_\Omega(v + \beta(u^k - u))$
    Evaluate $\zeta_k = \langle v + \beta(u^k - u), u^k - w^k \rangle$
    if $\zeta_k \leq \alpha$ **then**
      break
    end if
    Set $\theta_k = \min\{1, \zeta_k/(\beta\|w^k - u^k\|^2)\}$
    Update $u^{k+1} = u^k + \theta_k(w^k - u^k)$
  end for
  Set $u^+ = u^k$
end function

SPIDER-CGS: Convex finite-sum

We consider SPIDER-CGS with

$$K_t = \lceil 2^{t/2} \rceil \quad \text{for} \quad t = 1, 2, \ldots, T.$$

Furthermore, we choose the sampling parameters as

$$S_{t,k} = 9K_t s_{t,K_t}^2 \qquad \mathcal{Q}_t = [n]$$

CndG subsolver parameters as

$$\beta_{t,k} = \frac{3}{2}L\gamma_{t,k} \qquad \alpha_{t,k} = \frac{2LD^2}{(s_{t,k}+1)^2}$$

and the learning rate parameter as

$$\gamma_{t,k} = \frac{3}{s_{t,k}+2} \quad \text{where} \quad s_{t,k} = \sum_{\tau=1}^{t-1} K_\tau + k$$

Theorem 5. *Consider the convex finite-sum template, and suppose that the assumptions in Section 3 for this template hold. Then, estimate $y^{t,k}$ of SPIDER-CGS with the parameter choices described above satisfies*

$$\mathbb{E}[F(y^{t,k})] - F^\star = \mathcal{O}\left(\frac{LD^2}{s_{t,k}^2}\right)$$

Corollary 5. *The ifo and lmo complexities of SPIDER-CGS for achieving ϵ-solution in this template are as follows:*

$$\begin{aligned} \#(\textit{ifo}) &= \mathcal{O}\left(n \ln\left(\tfrac{LD^2}{\epsilon}\right) + \tfrac{L^2D^4}{\epsilon^2}\right) \\ \#(\textit{lmo}) &= \mathcal{O}(\tfrac{LD^2}{\epsilon}) \end{aligned}$$

Remark that the STORC (Hazan & Luo, 2016) has a better *ifo* complexity, but under the additional assumption of Lipschitz continuity of F.

SPIDER-CGS: Convex expectation minimization

We consider SPIDER-CGS with

$$K_t = \lceil 2^{t/2} \rceil \text{ for } t = 1, 2, \ldots, T.$$

Furthermore, we choose the sampling parameters as

$$S_{t,k} = 9K_t s_{t,K_t}^2 \qquad Q_t = \lceil \frac{\sigma^2 s_{t,K_t}^4}{L^2D^2} \rceil$$

CndG subsolver parameters as

$$\beta_{t,k} = \frac{3}{2}L\gamma_{t,k} \qquad \alpha_{t,k} = \frac{2LD^2}{(s_{t,k}+1)^2}$$

and the learning rate parameter as

$$\gamma_{t,k} = \frac{2}{s_{t,k}+1} \text{ where } s_{t,k} = \sum_{\tau=1}^{t-1} K_\tau + k$$

Theorem 6. *Consider the convex expectation minimization template, and suppose that the assumptions in Section 3 for this template hold. Then, estimate $y^{t,k}$ of SPIDER-CGS with the parameter choices described above satisfies*

$$\mathbb{E}[F(y^{t,k})] - F^\star = \mathcal{O}\left(\frac{LD^2}{s_{t,k}^2}\right)$$

Corollary 6. *The sfo and lmo complexities of SPIDER-CGS for achieving ϵ-solution in convex expectation minimization problems are as follows:*

$$\begin{aligned} \#(\textit{sfo}) &= \mathcal{O}\left(\tfrac{\sigma^2D^2 + L^2D^4}{\epsilon^2}\right) \\ \#(\textit{lmo}) &= \mathcal{O}(\tfrac{LD^2}{\epsilon}) \end{aligned}$$

SPIDER-CGS: Non-convex finite-sum

We consider SPIDER-CGS with

$$K_t = K = \lceil \sqrt{n} \rceil.$$

Furthermore, we choose the sampling parameters as

$$S_{t,k} = K \qquad \mathcal{Q}_t = [n]$$

CndG subsolver parameters as

$$\beta_{t,k} = \frac{3}{2}L\gamma \qquad \alpha_{t,k} = LD^2\gamma$$

and the learning rate parameter as

$$\gamma_{t,k} = \gamma = \frac{1}{\sqrt{s_{T,K}}} \text{ where } s_{T,K} = TK.$$

Theorem 7. *Consider the non-convex finite-sum template, and suppose that the assumptions in Section 3 for this template hold. Denote by y^{out} an iterate $y^{t,k}$ of SPIDER-CGS chosen uniformly random over all (t,k) pairs up to (T,K). Then, the following bound on the FW-gap holds:*

$$\mathbb{E}[\mathcal{G}(y^{out})] = \mathcal{O}\left(\frac{\mathcal{E} + LD^2}{\sqrt{s_{T,K}}}\right)$$

Corollary 7. *The ifo and lmo complexities of SPIDER-CGS for achieving ϵ-solution in non-convex finite-sum are*

$$\begin{aligned} \#(\textit{ifo}) &= \mathcal{O}\left(\tfrac{\sqrt{n}}{\epsilon^2}(\mathcal{E}^2 + L^2D^4)\right) \\ \#(\textit{lmo}) &= \mathcal{O}\left(\tfrac{1}{\epsilon^2}(\mathcal{E}^2 + L^2D^4)\right) \end{aligned}$$

SPIDER-CGS: Non-convex expectation minimization

We consider SPIDER-CGS with

$$K_t = K = \lceil \sigma/\epsilon \rceil.$$

Furthermore, we choose the sampling parameters as

$$S_{t,k} = K \qquad Q_t = \lceil 4(\sigma/\epsilon)^2 \rceil$$

CndG subsolver parameters as

$$\beta_{t,k} = \frac{3}{2}L\gamma \qquad \alpha_{t,k} = LD^2\gamma$$

and the learning rate parameter as

$$\gamma_{t,k} = \gamma = \frac{1}{\sqrt{s_{T,K}}} \text{ where } s_{T,K} = TK.$$

Theorem 8. *Consider the non-convex expectation minimization template, and suppose that the assumptions in Section 3 for this template hold. Denote by y^{out} an iterate $y^{t,k}$ of*

SPIDER-CGS chosen uniformly random over all (t, k) *pairs up to* (T, K)*. Then, the following bound holds:*

$$\mathbb{E}[\mathcal{G}(y^{out})] = \mathcal{O}\left(\frac{\mathcal{E} + LD^2}{\sqrt{s_{T,K}}}\right) + \frac{\epsilon}{2}$$

Corollary 8. *The sfo and lmo complexities of SPIDER-CGS for achieving* ϵ*-solution in non-convex expectation minimization problems are as follows:*

$$\begin{aligned} \#(\textit{sfo}) &= \mathcal{O}\left(\tfrac{\sigma}{\epsilon^3}(\mathcal{E}^2 + L^2D^4)\right) \\ \#(\textit{lmo}) &= \mathcal{O}\left(\tfrac{1}{\epsilon^2}(\mathcal{E}^2 + L^2D^4)\right) \end{aligned}$$

6. Comparison & Discussions

This section presents an extensive comparison of theoretical aspects of FW methods. Table 1 compiles a summary of this comparison.

6.1. Convex optimization camp

Batch setting. FW achieves an ϵ-solution after $\mathcal{O}(1/\epsilon)$ iterations. This complexity is optimal for a large class of methods that construct the decision variable through convex combination of *lmo* outputs (Lan, 2014). CGS, on the other side, enjoys $\mathcal{O}(1/\sqrt{\epsilon})$ first order oracle complexity while keeping the same $\mathcal{O}(1/\epsilon)$ *lmo* complexity, by reusing the same gradients over multiple iterations Lan & Zhou (2016).

Stochastic setting. Hazan & Kale (2012) propose Online-FW for an online-learning setting, but as mentioned later by Hazan & Luo (2016), these results can be translated to the stochastic template via standard conversion approaches, and gets $\mathcal{O}(1/\epsilon^4)$ and $\mathcal{O}(1/\epsilon^2)$ complexities for *sfo* and *lmo* calls respectively.

A natural extension of FW for stochastic setting is described by Hazan & Luo (2016), as shown in Algorithm 4. This method (SFW) is shown to converge with $\mathcal{O}(1/k)$ rate when the sample size $S_k = \Theta(k^2)$, hence it provides an ϵ-solution with $\mathcal{O}(1/\epsilon^3)$ *sfo* and $\mathcal{O}(1/\epsilon)$ *lmo* complexities.

Algorithm 4 Stochastic Frank-Wolfe

Input: $x^1 \in \Omega$
for $k = 1, 2, \ldots, K$ **do**
Draw S_k samples $\mathcal{S}_k$
Compute $w^k \in \mathrm{lmo}_\Omega\left(\nabla f_{\mathcal{S}_k}(x^k)\right)$
Update $x^{k+1} = x^k + \eta_k\,(w^k - x^k)$
end for

Lan & Zhou (2016) extend their CGS framework to the stochastic setting by introducing SCGS in Section 3 of their original work. While keeping the optimal $\mathcal{O}(1/\epsilon)$ *lmo* complexity, SCGS achieves $\mathcal{O}(1/\epsilon^2)$ *sfo* complexity, which even gets $\mathcal{O}(1/\epsilon)$ under strong convexity assumption.

Hazan & Luo (2016) introduce the stochastic variance reduced Frank-Wolfe method (SVRF) by adopting the variance reduction techniques from Johnson & Zhang (2013) and Mahdavi et al. (2013). SVRF is explicitly designed for the finite-sum setting, and it requires $\mathcal{O}(n\ln(1/\epsilon))$ full gradients as well as $\mathcal{O}(1/\epsilon^2)$ *ifo* and $\mathcal{O}(1/\epsilon)$ *lmo* to get an ϵ-solution.

To further improve *ifo* complexity of SVRF, Hazan & Luo (2016) design a variant based on CGS. This variant, stochastic variance reduced condition gradient sliding (STORC), also requires $\mathcal{O}(n\ln(1/\epsilon))$ full gradients and $\mathcal{O}(1/\epsilon)$ *lmo*, but it enjoys a reduced number of *ifo* calls at $\mathcal{O}(1/\epsilon^{1.5})$. Compared to SVRF, however, STORC additionally assumes that F is Lipschitz continuous in domain Ω. Also remark that STORC gets better rates under additional assumptions such as strong-convexity.

Lu & Freund (2018) propose a stochastic FW variant which requires $\mathcal{O}(1/\epsilon)$ *lmo* and $\mathcal{O}(n + 1/\epsilon)$ *ifo* complexity for the convex finite-sum. However, the proposed method relies on a special structure of the objective function, that f_i are univariate functions of the fitted value $\langle a_i, x\rangle$ for some given data sample a_i.

All stochastic FW variants we discussed up to know are based on an increasing mini-batch size. Very recently, Mokhtari et al. (2018) have proposed an alternative scheme (SFW-1) for expectation minimization setting, which requires a single *sfo* at each iteration. Nevertheless, SFW-1 has an arguably worse computational complexity compared to SFW, with its $\mathcal{O}(1/\epsilon^3)$ calls of *sfo* and *lmo*. We emphasize the applications of SFW-1 in submodular maximization, but this is beyond the scope of our work.

For the convex finite-sum setting, SPIDER-FW and SPIDER-CGS share the same complexities as SVRF.

6.2. Non-convex optimization camp

Batch setting. FW converges asymptotically to a stationary point, see Section 2.2 in (Bertsekas, 1999). To our knowledge, Yu et al. (2014) shows the first convergence rates for FW in non-convex setting, and Lacoste-Julien (2016) proves a non-asymptotic $\mathcal{O}(1/\sqrt{k})$ rate in FW-gap for a FW variant with line-search.

Stochastic setting. As shown by Reddi et al. (2016), SFW achieves to an ϵ-solution with $\mathcal{O}(1/\epsilon^4)$ *sfo* and $\mathcal{O}(1/\epsilon^2)$ *lmo* complexities. Moreover, they also analyze SVRF in the non-convex setting (but they call it SVFW), and prove that it takes $\mathcal{O}(1/\epsilon^{10/3})$ *sfo* and $\mathcal{O}(1/\epsilon^2)$ *lmo* complexity for this method to get an ϵ-solution. In the finite-sum setting, the former is replaced by $\mathcal{O}(n + n^{2/3}/\epsilon^2)$ *ifo* calls.

Reddi et al. (2016) also propose a variant (SAGAFW) based on the SAGA variance reduction technique described by

	convex				non-convex			
	finite-sum		expectation		finite-sum		expectation	
	(*ifo*)	(*lmo*)	(*sfo*)	(*lmo*)	(*ifo*)	(*lmo*)	(*sfo*)	(*lmo*)
FW	$\mathcal{O}(n\epsilon^{-1})$	$\mathcal{O}(\epsilon^{-1})$	-	-	$\mathcal{O}(n\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-2})$	-	-
CGS	$\mathcal{O}(n\epsilon^{-1/2})$	$\mathcal{O}(\epsilon^{-1})$	-	-	$\mathcal{O}(n\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-2})$	-	-
SFW	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-2})$
SFW-1	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-3})$	-	-	-	-
Online-FW	$\mathcal{O}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-2})$	-	-	-	-
SCGS	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-2})$
SVRF / SVFW	$\mathcal{O}(n\ln(\epsilon^{-1})+\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-1})$	-	-	$\mathcal{O}(n+n^{2/3}\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-10/3})$	$\mathcal{O}(\epsilon^{-2})$
STORC†	$\mathcal{O}(n\ln(\epsilon^{-1})+\epsilon^{-3/2})$	$\mathcal{O}(\epsilon^{-1})$	-	-	-	-	-	-
SPIDER-FW	$\mathcal{O}(n\ln(\epsilon^{-1})+\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(n^{1/2}\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-2})$
SPIDER-CGS	$\mathcal{O}(n\ln(\epsilon^{-1})+\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(n^{1/2}\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-2})$

Table 1: Comparison of conditional gradient methods for stochastic optimization. Contribution of *this work* is highlighted with blue font. See Section 6 for more details.
FW (Frank & Wolfe, 1956; Jaggi, 2013) , CGS (Lan & Zhou, 2016) , SFW (Hazan & Luo, 2016; Reddi et al., 2016) , SFW-1 (Mokhtari et al., 2018) , Online-FW (Hazan & Kale, 2012) , SCGS (Lan & Zhou, 2016) , SVRF / SVFW (Hazan & Luo, 2016; Reddi et al., 2016) , STORC (Hazan & Luo, 2016)

Defazio et al. (2014). However, we omit SAGAFW because there is an issue in the analysis of this method (while telescoping Eq.(14), in page 1249).

Qu et al. (2018) show the convergence rate for special instances of CGS and SCGS in the non-convex setting. However, they consider a different convergence criterion based on a proximal gradient mapping rather than the conventional FW-gap. Consequently, their results are incomparable with the rest of the literature. For the fact that we are running a *projection-free* method, the FW-gap is a more natural choice than the projection/proximal gradient norm.

We provide a parameter setting and a compact proof for CGS and SCGS in the supplemental material. Note however this setting simply gets the same guarantees as FW and SFW respectively. Whether or not CGS can provide improved oracle complexities compared to FW in the non-convex setting, is an open problem.

For the non-convex setting, SPIDER-FW and SPIDER-CGS have the same oracle complexities, superior to SVRF (which is the state-of-the-art to our knowledge) for finite-sum and expectation minimization problems.

6.3. Results from Concurrent Works

By the time we prepared this manuscript, the idea of combining SPIDER with the FW analysis was not explored yet. However, stochastic variance reduction methods and FW-type algorithms are both very active research fields. In this part, we discuss some results from a few concurrent works that appeared after we submitted our paper for review.

The recent work by Shen et al. (2019) is very closely related to our approach. They propose a class of methods based on the CGM and various variance reduction techniques for the non-convex finite-sum setting, including the SPIDER-FW. Besides, they also propose extensions that use second-order approximations. Finally, they provide simulation studies to compare empirical performance of different variants. We refer to this paper for a numerical comparison.

Hassani et al. (2019) introduce a novel variance reduced CGM method, but their work focuses primarily on the submodular maximization. Accordingly, they consider a more general expectation minimization template (the so-called non-oblivious setting) where the probability distribution depends on the decision variable x and may change during the optimization procedure. Therefore, the proposed method requires some further assumptions and modifications involving computations with the Hessian approximation. Finally, Zhang et al. (2019) consider a stochastic CGM approach with SPIDER in the distributed and quantized settings.

7. Concluding Remarks

We have proposed two novel FW-type methods based on the idea of blending the recent variance reduction technique SPIDER into FW and CGS frameworks. We have shown that the resulting methods enjoy superior oracle complexities in various convex and non-convex optimization templates. Extension of our framework for the strongly convex case is left open. Developing a well-tuned implementation, including one that incorporates parallel optimization, is an important piece of future work.

Acknowledgements

VC was supported by the Swiss National Science Foundation (SNSF) under grant number 200021_178865/1. VC has received funding for this project from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no 725594 - time-data). SS acknowledges support from an Amazon Research Award and the NSF-CAREER award (id 1846088). The authors thank Maria Vladarean for pointing out an error in the initial version of this work.

References

Agarwal, A. and Bottou, L. A lower bound for the optimization of finite sums. arXiv:1410.0723, 2014.

Bertsekas, D. *Nonlinear Programming*. Athena Scientific, 2 edition, 1999.

Defazio, A., Bach, F., and Lacoste-Julien, S. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems 28*, 2014.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems 31*, 2018.

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

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

Hazan, E. Sparse approximate solutions to semidefinite programs. In *Proc. 8th Latin American Conf. Theoretical Informatics*, pp. 306–316, 2008.

Hazan, E. and Kale, S. Projection–free online learning. In *Proc. 29th Int. Conf. Machine Learning*, 2012.

Hazan, E. and Luo, H. Variance-reduced and projection-free stochastic optimization. In *Proc. 33rd Int. Conf. Machine Learning*, 2016.

Jaggi, M. Revisiting Frank–Wolfe: Projection–free sparse convex optimization. In *Proc. 30th Int. Conf. Machine Learning*, 2013.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, 2013.

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

Lacoste-Julien, S., Jaggi, M., Schmidt, M., and Pletscher, P. Block-coordinate Frank-Wolfe optimization for structural SVMs. In *Proc. 30th Int. Conf. Machine Learning*, 2013.

Lan, G. The complexity of large–scale convex programming under a linear optimization oracle. arXiv:1309.5550v2, 2014.

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

Lu, H. and Freund, R. M. Generalized stochastic frank-wolfe algorithm with stochastic" substitute"gradient for structured convex optimization. arXiv:1807.07680, 2018.

Mahdavi, M., Zhang, L., and Jin, R. Mixed optimization for smooth functions. In *Advances in Neural Information Processing Systems 26*, 2013.

Mokhtari, A., Hassani, H., and Karbasi, A. Stochastic conditional gradient methods: From convex minimization to submodular maximization. arXiv:1804.09554, 2018.

Nemirovski, A. and Yudin, D. *Problem Complexity and Method Efficiency in Optimization*. John Wiley & Sons, 1983.

Nesterov, Y. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takác, M. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *Proc. 34th Int. Conf. Machine Learning*, 2017.

Qu, C., Li, Y., and Xu, H. Non-convex conditional gradient sliding. In *Proc. 35th Int. Conf. Machine Learning*, 2018.

Ravi, S. N., Dinh, T., Lokhande, V. S. R., and Singh, V. Constrained deep learning using conditional gradient and applications in computer vision. arXiv:1803.06453v1, 2018.

Reddi, S. J., Sra, S., Póczos, B., and Smola, A. Stochastic Frank-Wolfe methods for nonconvex optimization. In *54th Annual Allerton Conf. Communication, Control, and Computing*, pp. 1244–1251, 2016.

Roux, N. L., Schmidt, M., and Bach, F. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems 26*, 2012.

Shen, Z., Fang, C., Zhao, P., Huang, J., and Qian, H. Complexities in projection-free stochastic non-convex minimization. In *Proc. 22nd Int. Conf. Artificial Intelligence and Statistics*, 2019.

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

Yu, Y., Zhang, X., and Schuurmans, D. Generalized conditional gradient for sparse estimation. arXiv:1410.4828v1, 2014.

Zhang, M., Chen, L., Mokhtari, A., Hassani, H., and Karbasi, A. Quantized frank-wolfe: Communication-efficient distributed optimization. arXiv:1902.06332, 2019.