Variance-Reduced and Projection-Free Stochastic Optimization

Elad Hazan EHAZAN@CS.PRINCETON.EDU
Princeton University, Princeton, NJ 08540, USA

Haipeng Luo HAIPENGL@CS.PRINCETON.EDU
Princeton University, Princeton, NJ 08540, USA

Abstract

The Frank-Wolfe optimization algorithm has recently regained popularity for machine learning applications due to its projection-free property and its ability to handle structured constraints. However, in the stochastic learning setting, it is still relatively understudied compared to the gradient descent counterpart. In this work, leveraging a recent variance reduction technique, we propose two stochastic Frank-Wolfe variants which substantially improve previous results in terms of the number of stochastic gradient evaluations needed to achieve $1-\epsilon$ accuracy. For example, we improve from $\mathcal{O}(\frac{1}{\epsilon})$ to $\mathcal{O}(\ln\frac{1}{\epsilon})$ if the objective function is smooth and strongly convex, and from $\mathcal{O}(\frac{1}{\epsilon^2})$ to $\mathcal{O}(\frac{1}{\epsilon^{1.5}})$ if the objective function is smooth and Lipschitz. The theoretical improvement is also observed in experiments on real-world datasets for a multiclass classification application.

1. Introduction

We consider the following optimization problem

$$\min_{\boldsymbol{w}\in\Omega} f(\boldsymbol{w}) = \min_{\boldsymbol{w}\in\Omega} \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{w})$$

which is an extremely common objective in machine learning. We are interested in the case where 1) n, usually corresponding to the number of training examples, is very large and therefore stochastic optimization is much more efficient; and 2) the domain Ω admits fast linear optimization, while projecting onto it is much slower, necessitating projection-free optimization algorithms. Examples of such problem include multiclass classification, multitask learning, recommendation systems, matrix learning and many more (see for example (Hazan & Kale, 2012; Hazan et al., 2012; Jaggi, 2013; Dudik et al., 2012; Zhang et al., 2012; Harchaoui et al., 2015)).

The Frank-Wolfe algorithm (Frank & Wolfe, 1956) (also known as *conditional gradient*) and it variants are natural candidates for solving these problems, due to its projection-free property and its ability to handle structured constraints. However, despite gaining more popularity recently, its applicability and efficiency in the stochastic learning setting, where computing stochastic gradients is much faster than computing exact gradients, is still relatively understudied compared to variants of projected gradient descent methods.

In this work, we thus try to answer the following question: *what running time can a projection-free algorithm achieve in terms of the number of stochastic gradient evaluations and the number of linear optimizations needed to achieve a certain accuracy?* Utilizing Nesterov's acceleration technique (Nesterov, 1983) and the recent variance reduction idea (Johnson & Zhang, 2013; Mahdavi et al., 2013), we propose two new algorithms that are substantially faster than previous work. Specifically, to achieve $1-\epsilon$ accuracy, while the number of linear optimization is the same as previous work, the improvement of the number of stochastic gradient evaluations is summarized in Table 1:

	previous work	**this work**
Smooth	$\mathcal{O}(\frac{1}{\epsilon^2})$	$\mathcal{O}(\frac{1}{\epsilon^{1.5}})$
Smooth and Strongly Convex	$\mathcal{O}(\frac{1}{\epsilon})$	$\mathcal{O}(\ln\frac{1}{\epsilon})$

Table 1: Comparisons of number of stochastic gradients

The extra overhead of our algorithms is computing at most $\mathcal{O}(\ln\frac{1}{\epsilon})$ exact gradients, which is computationally insignificant compared to the other operations. A more detailed comparisons to previous work is included in Table 2, which will be further explained in Section 2.

While the idea of our algorithms is quite straightforward,

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

we emphasize that our analysis is non-trivial, especially for the second algorithm where the convergence of a sequence of auxiliary points in Nesterov's algorithm needs to be shown.

To support our theoretical results, we also conducted experiments on three large real-word datasets for a multiclass classification application. These experiments show significant improvement over both previous projection-free algorithms and algorithms such as projected stochastic gradient descent and its variance-reduced version.

The rest of the paper is organized as follows: Section 2 setups the problem more formally and discusses related work. Our two new algorithms are presented and analyzed in Section 3 and 4, followed by experiment details in Section 5.

2. Preliminary and Related Work

We assume each function f_i is convex and L-smooth, that is, for any $\boldsymbol{w}, \boldsymbol{v} \in \Omega$,

$$\begin{aligned}\nabla f_i(\boldsymbol{v})^\top(\boldsymbol{w}-\boldsymbol{v}) &\le f_i(\boldsymbol{w}) - f_i(\boldsymbol{v}) \\ &\le \nabla f_i(\boldsymbol{v})^\top(\boldsymbol{w}-\boldsymbol{v}) + \frac{L}{2}\|\boldsymbol{w}-\boldsymbol{v}\|^2 .\end{aligned}$$

We will use two more important properties of smoothness. The first one is

$$\begin{aligned}&\|\nabla f_i(\boldsymbol{w}) - \nabla f_i(\boldsymbol{v})\|^2 \le \\ &\qquad 2L(f_i(\boldsymbol{w}) - f_i(\boldsymbol{v}) - \nabla f_i(\boldsymbol{v})^\top(\boldsymbol{w}-\boldsymbol{v}))\end{aligned} \tag{1}$$

(proven in Appendix A for completeness), and the second one is

$$\begin{aligned}&f_i(\lambda\boldsymbol{w} + (1-\lambda)\boldsymbol{v}) \ge \\ &\qquad \lambda f_i(\boldsymbol{w}) + (1-\lambda) f_i(\boldsymbol{v}) - \frac{L}{2}\lambda(1-\lambda)\|\boldsymbol{w}-\boldsymbol{v}\|^2\end{aligned} \tag{2}$$

for any $\boldsymbol{w}, \boldsymbol{v} \in \Omega$ and $\lambda \in [0,1]$. Notice that $f = \frac{1}{n}\sum_{i=1}^n f_i$ is also L-smooth since smoothness is preserved under convex combinations.

For some cases, we also assume each f_i is G-Lipschitz: $\|\nabla f_i(\boldsymbol{w})\| \le G$ for any $\boldsymbol{w} \in \Omega$, and f (although not necessarily each f_i) is α-strongly convex, that is,

$$f(\boldsymbol{w}) - f(\boldsymbol{v}) \le \nabla f(\boldsymbol{w})^\top(\boldsymbol{w}-\boldsymbol{v}) - \frac{\alpha}{2}\|\boldsymbol{w}-\boldsymbol{v}\|^2$$

for any $\boldsymbol{w}, \boldsymbol{v} \in \Omega$. As usual, $\mu = \frac{L}{\alpha}$ is called the condition number of f.

We assume the domain $\Omega \in \mathbb{R}^d$ is a compact convex set with diameter D. We are interested in the case where linear optimization on Ω, formally $\operatorname{argmin}_{\boldsymbol{v}\in\Omega} \boldsymbol{w}^\top\boldsymbol{v}$ for any $\boldsymbol{w} \in \mathbb{R}^d$, is much faster than projection onto Ω, formally $\operatorname{argmin}_{\boldsymbol{v}\in\Omega}\|\boldsymbol{w}-\boldsymbol{v}\|^2$. Examples of such domains include the set of all bounded trace norm matrices, the convex hull of all rotation matrices, flow polytope and many more (see for instance (Hazan & Kale, 2012)).

2.1. Example Application: Multiclass Classification

Consider a multiclass classification problem where a set of training examples $(\boldsymbol{e}_i, y_i)_{i=1,\dots,n}$ is given beforehand. Here $\boldsymbol{e}_i \in \mathbb{R}^m$ is a feature vector and $y_i \in \{1,\dots,h\}$ is the label. Our goal is to find an accurate linear predictor, a matrix $\boldsymbol{w} = [\boldsymbol{w}_1^\top;\dots,\boldsymbol{w}_h^\top] \in \mathbb{R}^{h\times m}$ that predicts $\operatorname{argmax}_\ell \boldsymbol{w}_\ell^\top\boldsymbol{e}$ for any example $\boldsymbol{e}$. Note that here the dimensionality d is hm.

Previous work (Dudik et al., 2012; Zhang et al., 2012) found that finding $\boldsymbol{w}$ by minimizing a regularized multivariate logistic loss gives a very accurate predictor in general. Specifically, the objective can be written in our notation with

$$f_i(\boldsymbol{w}) = \log\left(1 + \sum_{\ell\neq y_i}\exp(\boldsymbol{w}_\ell^\top\boldsymbol{e}_i - \boldsymbol{w}_{y_i}^\top\boldsymbol{e}_i)\right)$$

and $\Omega = \{\boldsymbol{w} \in \mathbb{R}^{h\times m} : \|\boldsymbol{w}\|_* \le \tau\}$ where $\|\cdot\|_*$ denotes the matrix trace norm. In this case, projecting onto Ω is equivalent to performing an SVD, which takes $\mathcal{O}(hm\min\{h,m\})$ time, while linear optimization on Ω amounts to finding the top singular vector, which can be done in time linear to the number of non-zeros in the corresponding h by m matrix, and is thus much faster. One can also verify that each f_i is smooth. The number of examples n can be prohibitively large for non-stochastic methods (for instance, tens of millions for the ImageNet dataset (Deng et al., 2009)), which makes stochastic optimization necessary.

2.2. Detailed Efficiency Comparisons

We call $\nabla f_i(\boldsymbol{w})$ a *stochastic gradient* for f at some $\boldsymbol{w}$, where i is picked from $\{1,\dots,n\}$ uniformly at random. Note that a stochastic gradient $\nabla f_i(\boldsymbol{w})$ is an unbiased estimator of the *exact gradient* $\nabla f(\boldsymbol{w})$. The efficiency of a projection-free algorithm is measured by how many numbers of exact gradient evaluations, stochastic gradient evaluations and linear optimizations respectively are needed to achieve $1-\epsilon$ accuracy, that is, to output a point $\boldsymbol{w} \in \Omega$ such that $\mathbb{E}[f(\boldsymbol{w}) - f(\boldsymbol{w}^*)] \le \epsilon$ where $\boldsymbol{w}^* \in \operatorname{argmin}_{\boldsymbol{w}\in\Omega} f(\boldsymbol{w})$ is any optimum.

In Table 2, we summarize the efficiency (and extra assumptions needed beside convexity and smoothness[1]) of existing algorithms in the literature as well as the two new algorithms we propose. Below we briefly explain these results from top to bottom.

[1] In general, condition "G-Lipschitz" in Table 2 means each f_i is G-Lipschitz, except for our STORC algorithm which only requires f being G-Lipschitz.

Algorithm	Extra Conditions	#Exact Gradients	#Stochastic Gradients	#Linear Optimizations
Frank-Wolfe		$\mathcal{O}(\frac{LD^2}{\epsilon})$	0	$\mathcal{O}(\frac{LD^2}{\epsilon})$
(Garber & Hazan, 2013)	α-strongly convex Ω is polytope	$\mathcal{O}(d\mu\rho\ln\frac{LD^2}{\epsilon})$	0	$\mathcal{O}(d\mu\rho\ln\frac{LD^2}{\epsilon})$
SFW	G-Lipschitz	0	$\mathcal{O}(\frac{G^2LD^4}{\epsilon^3})$	$\mathcal{O}(\frac{LD^2}{\epsilon})$
Online-FW (Hazan & Kale, 2012)	G-Lipschitz	0	$\mathcal{O}(\frac{d^2(LD^2+GD)^4}{\epsilon^4})$	$\mathcal{O}(\frac{d(LD^2+GD)^2}{\epsilon^2})$
	G-Lipschitz ($L=\infty$ allowed)	0	$\mathcal{O}(\frac{G^4D^4}{\epsilon^4})$	$\mathcal{O}(\frac{G^4D^4}{\epsilon^4})$
SCGS (Lan & Zhou, 2014)	G-Lipschitz	0	$\mathcal{O}(\frac{G^2D^2}{\epsilon^2})$	$\mathcal{O}(\frac{LD^2}{\epsilon})$
	G-Lipschitz α-strongly convex	0	$\mathcal{O}(\frac{G^2}{\alpha\epsilon})$	$\mathcal{O}(\frac{LD^2}{\epsilon})$
SVRF (this work)		$\mathcal{O}(\ln\frac{LD^2}{\epsilon})$	$\mathcal{O}(\frac{L^2D^4}{\epsilon^2})$	$\mathcal{O}(\frac{LD^2}{\epsilon})$
STORC (this work)	G-Lipschitz	$\mathcal{O}(\ln\frac{LD^2}{\epsilon})$	$\mathcal{O}(\frac{\sqrt{L}D^2G}{\epsilon^{1.5}})$	$\mathcal{O}(\frac{LD^2}{\epsilon})$
	$\nabla f(\boldsymbol{w}^*)=\mathbf{0}$	$\mathcal{O}(\ln\frac{LD^2}{\epsilon})$	$\mathcal{O}(\frac{LD^2}{\epsilon})$	$\mathcal{O}(\frac{LD^2}{\epsilon})$
	α-strongly convex	$\mathcal{O}(\ln\frac{LD^2}{\epsilon})$	$\mathcal{O}(\mu^2\ln\frac{LD^2}{\epsilon})$	$\mathcal{O}(\frac{LD^2}{\epsilon})$

Table 2: Comparisons of different Frank-Wolfe variants (see Section 2.2 for further explanations).

The standard Frank-Wolfe algorithm:

$$\begin{aligned}\boldsymbol{v}_k &= \operatorname*{argmin}_{\boldsymbol{v}\in\Omega} \nabla f(\boldsymbol{w}_{k-1})^\top \boldsymbol{v} \\ \boldsymbol{w}_k &= (1-\gamma_k)\boldsymbol{w}_{k-1} + \gamma_k \boldsymbol{v}_k\end{aligned} \tag{3}$$

for some appropriate chosen γ_k requires $\mathcal{O}(\frac{1}{\epsilon})$ iteration without additional conditions (Frank & Wolfe, 1956; Jaggi, 2013). In a recent paper, Garber & Hazan (2013) give a variant that requires $\mathcal{O}(d\mu\rho\ln\frac{1}{\epsilon})$ iterations when f is strongly convex and smooth, and Ω is a polytope[2]. Although the dependence on ϵ is much better, the geometric constant ρ depends on the polyhedral set and can be very large. Moreover, each iteration of the algorithm requires further computation besides the linear optimization step.

The most obvious way to obtain a stochastic Frank-Wolfe variant is to replace $\nabla f(\boldsymbol{w}_{k-1})$ by some $\nabla f_i(\boldsymbol{w}_{k-1})$, or more generally the average of some number of iid samples of $\nabla f_i(\boldsymbol{w}_{k-1})$ (mini-batch approach). We call this method SFW and include its analysis in Appendix B since we do not find it explicitly analyzed before. SFW needs $\mathcal{O}(\frac{1}{\epsilon^3})$ stochastic gradients and $\mathcal{O}(\frac{1}{\epsilon})$ linear optimization steps to reach an ϵ-approximate optimum.

The work by Hazan & Kale (2012) focuses on a online learning setting. One can extract two results from this work for the setting studied here.[3] In any case, the result is worse than SFW for both the number of stochastic gradients and the number of linear optimizations.

Stochastic Condition Gradient Sliding (SCGS), recently proposed by (Lan & Zhou, 2014), uses Nesterov's acceleration technique to speed up Frank-Wolfe. Without strong convexity, SCGS needs $\mathcal{O}(\frac{1}{\epsilon^2})$ stochastic gradients, improving SFW. With strong convexity, this number can even be improved to $\mathcal{O}(\frac{1}{\epsilon})$. In both cases, the number of linear optimization steps is $\mathcal{O}(\frac{1}{\epsilon})$.

The key idea of our algorithms is to combine the variance reduction technique proposed in (Johnson & Zhang, 2013; Mahdavi et al., 2013) with some of the above-mentioned algorithms. For example, our algorithm SVRF combines this technique with SFW, also improving the number of stochastic gradients from $\mathcal{O}(\frac{1}{\epsilon^3})$ to $\mathcal{O}(\frac{1}{\epsilon^2})$, but without any extra conditions (such as Lipschitzness required for SCGS). More importantly, despite having seemingly same convergence rate, SVRF substantially outperforms SCGS empirically (see Section 5).

On the other hand, our second algorithm STORC combines variance reduction with SCGS, providing even further improvements. Specifically, the number of stochastic gradients is improved to: $\mathcal{O}(\frac{1}{\epsilon^{1.5}})$ when f is Lipschitz; $\mathcal{O}(\frac{1}{\epsilon})$ when $\nabla f(\boldsymbol{w}^*)=\mathbf{0}$; and finally $\mathcal{O}(\ln\frac{1}{\epsilon})$ when f is strongly convex. Note that the condition $\nabla f(\boldsymbol{w}^*)=\mathbf{0}$ essentially means that $\boldsymbol{w}^*$ is in the interior of Ω, but it is still an interesting case when the optimum is not unique and doing unconstraint optimization would not necessary return a point in Ω.

[2] See also recent follow up work (Lacoste-Julien & Jaggi, 2015).

[3] The first result comes from the setting where the online loss functions are stochastic, and the second one comes from a completely online setting with the standard online-to-batch conversion.

Both of our algorithms require $\mathcal{O}(\frac{1}{\epsilon})$ linear optimization steps as previous work, and overall require computing $\mathcal{O}(\ln \frac{LD^2}{\epsilon})$ exact gradients. However, we emphasize that this extra overhead is much more affordable compared to non-stochastic Frank-Wolfe (that is, computing exact gradients every iteration) since it does not have any polynomial dependence on parameters such as d, L or μ.

2.3. Variance-Reduced Stochastic Gradients

Originally proposed in (Johnson & Zhang, 2013) and independently in (Mahdavi et al., 2013), the idea of variance-reduced stochastic gradients is proven to be highly useful and has been extended to various different algorithms (such as (Frostig et al., 2015; Moritz et al., 2016)).

A variance-reduced stochastic gradient at some point $\boldsymbol{w} \in \Omega$ with some *snapshot* $\boldsymbol{w}_0 \in \Omega$ is defined as

$$\tilde{\nabla} f(\boldsymbol{w}; \boldsymbol{w}_0) = \nabla f_i(\boldsymbol{w}) - (\nabla f_i(\boldsymbol{w}_0) - \nabla f(\boldsymbol{w}_0)),$$

where i is again picked from $\{1, \dots, n\}$ uniformly at random. The snapshot $\boldsymbol{w}_0$ is usually a decision point from some previous iteration of the algorithm and its exact gradient $\nabla f(\boldsymbol{w}_0)$ has been pre-computed before, so that computing $\tilde{\nabla} f(\boldsymbol{w}; \boldsymbol{w}_0)$ only requires two standard stochastic gradient evaluations: $\nabla f_i(\boldsymbol{w})$ and $\nabla f_i(\boldsymbol{w}_0)$.

A variance-reduced stochastic gradient is clearly also unbiased, that is, $\mathbb{E}[\tilde{\nabla} f(\boldsymbol{w}; \boldsymbol{w}_0)] = \nabla f(\boldsymbol{w})$. More importantly, the term $\nabla f_i(\boldsymbol{w}_0) - \nabla f(\boldsymbol{w}_0)$ serves as a correction term to reduce the variance of the stochastic gradient. Formally, one can prove the following

Lemma 1. *For any $\boldsymbol{w}, \boldsymbol{w}_0 \in \Omega$, we have*

$$\mathbb{E}[\|\tilde{\nabla} f(\boldsymbol{w}; \boldsymbol{w}_0) - \nabla f(\boldsymbol{w})\|^2] \\ \leq 6L(2\mathbb{E}[f(\boldsymbol{w}) - f(\boldsymbol{w}^*)] + \mathbb{E}[f(\boldsymbol{w}_0) - f(\boldsymbol{w}^*)]).$$

In words, the variance of the variance-reduced stochastic gradient is bounded by how close the current point and the snapshot are to the optimum. The original work proves a bound on $\mathbb{E}[\|\tilde{\nabla} f(\boldsymbol{w}; \boldsymbol{w}_0)\|^2]$ under the assumption $\nabla f(\boldsymbol{w}^*) = \mathbf{0}$, which we do not require here. However, the main idea of the proof is similar and we defer it to Section 6.

3. Stochastic Variance-Reduced Frank-Wolfe

With the previous discussion, our first algorithm is pretty straightforward: compared to the standard Frank-Wolfe, we simply replace the exact gradient with the average of a mini-batch of variance-reduced stochastic gradients, and take snapshots every once in a while. We call this algorithm Stochastic Variance-Reduced Frank-Wolfe (SVRF), whose pseudocode is presented in Alg 1. The convergence rate of this algorithm is shown in the following theorem.

Algorithm 1 Stochastic Variance-Reduced Frank-Wolfe (SVRF)

Input: Objective function $f = \frac{1}{n}\sum_{i=1}^n f_i$.
Input: Parameters γ_k, m_k and N_k.
Initialize: $\boldsymbol{w}_0 = \min_{\boldsymbol{w} \in \Omega} \nabla f(\boldsymbol{x})^\top \boldsymbol{w}$ for some arbitrary $\boldsymbol{x} \in \Omega$.
for $t = 1, 2, \dots, T$ **do**
  Take snapshot: $\boldsymbol{x}_0 = \boldsymbol{w}_{t-1}$ and compute $\nabla f(\boldsymbol{x}_0)$.
  for $k = 1$ **to** N_t **do**
    Compute $\tilde{\nabla}_k$, the average of m_k iid samples of $\tilde{\nabla} f(\boldsymbol{x}_{k-1}, \boldsymbol{x}_0)$.
    Compute $\boldsymbol{v}_k = \min_{\boldsymbol{v} \in \Omega} \tilde{\nabla}_k^\top \boldsymbol{v}$.
    Compute $\boldsymbol{x}_k = (1 - \gamma_k)\boldsymbol{x}_{k-1} + \gamma_k \boldsymbol{v}_k$.
  end for
  Set $\boldsymbol{w}_t = \boldsymbol{x}_{N_t}$.
end for

Theorem 1. *With the following parameters,*

$$\gamma_k = \frac{2}{k+1}, \; m_k = 96(k+1), \; N_t = 2^{t+3} - 2,$$

Algorithm 1 ensures $\mathbb{E}[f(\boldsymbol{w}_t) - f(\boldsymbol{w}^*)] \leq \frac{LD^2}{2^{t+1}}$ for any t.

Before proving this theorem, we first show a direct implication of this convergence result.

Corollary 1. *To achieve $1 - \epsilon$ accuracy, Algorithm 1 requires $\mathcal{O}(\ln \frac{LD^2}{\epsilon})$ exact gradient evaluations, $\mathcal{O}(\frac{L^2D^4}{\epsilon^2})$ stochastic gradient evaluations and $\mathcal{O}(\frac{LD^2}{\epsilon})$ linear optimizations.*

Proof. According to the algorithm and the choice of parameters, it is clear that these three numbers are $T + 1$, $\sum_{t=1}^T \sum_{k=1}^{N_t} m_k = \mathcal{O}(4^T)$ and $\sum_{t=1}^T N_t = \mathcal{O}(2^T)$ respectively. Theorem 1 implies that T should be of order $\Theta(\log_2 \frac{LD^2}{\epsilon})$. Plugging in all parameters concludes the proof. □

To prove Theorem 1, we first consider a fixed iteration t and prove the following lemma:

Lemma 2. *For any k, we have*

$$\mathbb{E}[f(\boldsymbol{x}_k) - f(\boldsymbol{w}^*)] \leq \frac{4LD^2}{k+2}$$

if $\mathbb{E}[\|\tilde{\nabla}_s - \nabla f(\boldsymbol{x}_{s-1})\|^2] \leq \frac{L^2D^2}{(s+1)^2}$ for all $s \leq k$.

We defer the proof of this lemma to Section 6 for coherence. With the help of Lemma 2, we are now ready to prove the main convergence result.

Proof of Theorem 1. We prove by induction. For $t = 0$, by smoothness, the optimality of $\boldsymbol{w}_0$ and convexity, we have

$$\begin{aligned} f(\boldsymbol{w}_0) &\le f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{w}_0 - \boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{w}_0 - \boldsymbol{x}\|^2 \\ &\le f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{w}^* - \boldsymbol{x}) + \frac{LD^2}{2} \\ &\le f(\boldsymbol{w}^*) + \frac{LD^2}{2}. \end{aligned}$$

Now assuming $\mathbb{E}[f(\boldsymbol{w}_{t-1}) - f(\boldsymbol{w}^*)] \le \frac{LD^2}{2^t}$, we consider iteration t of the algorithm and use another induction to show $\mathbb{E}[f(\boldsymbol{x}_k) - f(\boldsymbol{w}^*)] \le \frac{4LD^2}{k+2}$ for any $k \le N_t$. The base case is trivial since $\boldsymbol{x}_0 = \boldsymbol{w}_{t-1}$. Suppose $\mathbb{E}[f(\boldsymbol{x}_{s-1}) - f(\boldsymbol{w}^*)] \le \frac{4LD^2}{s+1}$ for any $s \le k$. Now because $\tilde{\nabla}_s$ is the average of m_s iid samples of $\tilde{\nabla} f(\boldsymbol{x}_{s-1}; \boldsymbol{x}_0)$, its variance is reduced by a factor of m_s. That is, with Lemma 1 we have

$$\begin{aligned} &\mathbb{E}[\|\tilde{\nabla}_s - \nabla f(\boldsymbol{x}_{s-1})\|^2] \\ &\le \frac{6L}{m_s}(2\mathbb{E}[f(\boldsymbol{x}_{s-1}) - f(\boldsymbol{w}^*)] + \mathbb{E}[f(\boldsymbol{x}_0) - f(\boldsymbol{w}^*)]) \\ &\le \frac{6L}{m_s}\left(\frac{8LD^2}{s+1} + \frac{LD^2}{2^t}\right) \\ &\le \frac{6L}{m_s}\left(\frac{8LD^2}{s+1} + \frac{8LD^2}{s+1}\right) = \frac{L^2D^2}{(s+1)^2}, \end{aligned}$$

where the last inequality is by the fact $s \le N_t = 2^{t+3} - 2$ and the last equality is by plugging the choice of m_s. Therefore the condition of Lemma 2 is satisfied and the induction is completed. Finally with the choice of N_t we thus prove $\mathbb{E}[f(\boldsymbol{w}_t) - f(\boldsymbol{w}^*)] = \mathbb{E}[f(\boldsymbol{x}_{N_t}) - f(\boldsymbol{w}^*)] \le \frac{4LD^2}{N_t+2} = \frac{LD^2}{2^{t+1}}$. □

We remark that in Alg 1, we essentially restart the algorithm (that is, reseting k to 1) after taking a new snapshot. However, another option is to continue increasing k and never reset it. Although one can show that this only leads to constant speed up for the convergence, it provides more stable update and is thus what we implement in experiments.

4. Stochastic Variance-Reduced Conditional Gradient Sliding

Our second algorithm applies variance reduction to the SCGS algorithm (Lan & Zhou, 2014). Again, the key difference is that we replace the stochastic gradients with the average of a mini-batch of variance-reduced stochastic gradients, and take snapshots every once in a while. See pseudocode in Alg 2 for details.

The algorithm makes use of two auxiliary sequences $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$ (Line 8 and 12), which is standard for Nesterov's algorithm. $\boldsymbol{x}_k$ is obtained by approximately solving a square norm regularized linear optimization so that it is close to $\boldsymbol{x}_{k-1}$ (Line 11). Note that this step does not require computing any extra gradients of f or f_i, and is done by performing the standard Frank-Wolfe algorithm (Eq. (3)) until the duality gap is at most a certain value $\eta_{t,k}$. The duality gap is a certificate of approximate optimality (see (Jaggi, 2013)), and is a side product of the linear optimization performed at each step, requiring no extra cost.

Algorithm 2 STOchastic variance-Reduced Conditional gradient sliding (STORC)

1: **Input:** Objective function $f = \frac{1}{n}\sum_{i=1}^n f_i$.
2: **Input:** Parameters γ_k, β_k, $\eta_{t,k}$, $m_{t,k}$ and N_t.
3: **Initialize:** $\boldsymbol{w}_0 = \min_{\boldsymbol{w}\in\Omega} \nabla f(\boldsymbol{x})^\top \boldsymbol{w}$ for some arbitrary $\boldsymbol{x} \in \Omega$.
4: **for** $t = 1, 2, \ldots$ **do**
5: Take snapshot: $\boldsymbol{y}_0 = \boldsymbol{w}_{t-1}$ and compute $\nabla f(\boldsymbol{y}_0)$.
6: Initialize $\boldsymbol{x}_0 = \boldsymbol{y}_0$.
7: **for** $k = 1$ **to** N_t **do**
8: Compute $\boldsymbol{z}_k = (1-\gamma_k)\boldsymbol{y}_{k-1} + \gamma_k \boldsymbol{x}_{k-1}$.
9: Compute $\tilde{\nabla}_k$, the average of $m_{t,k}$ iid samples of $\tilde{\nabla} f(\boldsymbol{z}_k; \boldsymbol{y}_0)$.
10: Let $g(\boldsymbol{x}) = \frac{\beta_k}{2}\|\boldsymbol{x} - \boldsymbol{x}_{k-1}\|^2 + \tilde{\nabla}_k^\top \boldsymbol{x}$.
11: Compute $\boldsymbol{x}_k$, the output of using standard Frank-Wolfe to solve $\min_{\boldsymbol{x}\in\Omega} g(\boldsymbol{x})$ until the duality gap is at most $\eta_{t,k}$, that is,

$$\max_{\boldsymbol{x}\in\Omega} \nabla g(\boldsymbol{x}_k)^\top (\boldsymbol{x}_k - \boldsymbol{x}) \le \eta_{t,k}\,. \qquad (4)$$

12: Compute $\boldsymbol{y}_k = (1-\gamma_k)\boldsymbol{y}_{k-1} + \gamma_k \boldsymbol{x}_k$.
13: **end for**
14: Set $\boldsymbol{w}_t = \boldsymbol{y}_{N_t}$.
15: **end for**

Also note that the stochastic gradients are computed at the sequence $\boldsymbol{z}_k$ instead of $\boldsymbol{y}_k$, which is also standard in Nesterov's algorithm. However, according to Lemma 1, we thus need to show the convergence rate of the auxiliary sequence $\boldsymbol{z}_k$, which appears to be rarely studied previously to the best our knowledge. This is one of the key steps in our analysis.

The main convergence result of STORC is the following:

Theorem 2. *With the following parameters (where D_t is defined later below):*

$$\gamma_k = \frac{2}{k+1},\ \beta_k = \frac{3L}{k},\ \eta_{t,k} = \frac{2LD_t^2}{N_t k},$$

Algorithm 2 ensures $\mathbb{E}[f(\boldsymbol{w}_t) - f(\boldsymbol{w}^*)] \le \frac{LD^2}{2^{t+1}}$ for any t if any of the following three cases holds:

(a) $\nabla f(\boldsymbol{w}^*) = \boldsymbol{0}$ and $D_t = D, N_t = \lceil 2^{\frac{t}{2}+2} \rceil, m_{t,k} = 900 N_t$.

(b) f is G-Lipschitz and $D_t = D, N_t = \lceil 2^{\frac{t}{2}+2} \rceil, m_{t,k} =$

$700N_t + \frac{24N_tG(k+1)}{LD}$.

(c) *f is α-strongly convex and $D_t^2 = \frac{\mu D^2}{2^{t-1}}, N_t = \lceil\sqrt{32\mu}\rceil, m_{t,k} = 5600N_t\mu$ where $\mu = \frac{L}{\alpha}$.*

Again we first give a direct implication of the above result:

Corollary 2. *To achieve $1-\epsilon$ accuracy, Algorithm 2 requires $\mathcal{O}(\ln\frac{LD^2}{\epsilon})$ exact gradient evaluations and $\mathcal{O}(\frac{LD^2}{\epsilon})$ linear optimizations. The numbers of stochastic gradient evaluations for Case (a), (b) and (c) are respectively $\mathcal{O}(\frac{LD^2}{\epsilon})$, $\mathcal{O}(\frac{LD^2}{\epsilon} + \frac{\sqrt{L}D^2G}{\epsilon^{1.5}})$ and $\mathcal{O}(\mu^2\ln\frac{LD^2}{\epsilon})$.*

Proof. Line 11 requires $\mathcal{O}(\frac{\beta_kD^2}{\eta_{t,k}})$ iterations of the standard Frank-Wolfe algorithm since $g(\boldsymbol{x})$ is β_k-smooth (see e.g. (Jaggi, 2013, Theorem 2)). So the numbers of exact gradient evaluations, stochastic gradient evaluations and linear optimizations are respectively $T+1$, $\sum_{t=1}^T\sum_{k=1}^{N_t} m_{t,k}$ and $\mathcal{O}(\sum_{t=1}^T\sum_{k=1}^{N_t}\frac{\beta_kD^2}{\eta_{t,k}})$. Theorem 2 implies that T should be of order $\Theta(\log_2\frac{LD^2}{\epsilon})$. Plugging in all parameters proves the corollary. □

To prove Theorem 2, we again first consider a fixed iteration t and use the following lemma, which is essentially proven in (Lan & Zhou, 2014). We include a distilled proof in Appendix C for completeness.

Lemma 3. *Suppose $\mathbb{E}[\|\boldsymbol{y}_0 - \boldsymbol{w}^*\|^2] \le D_t^2$ holds for some positive constant $D_t \le D$. Then for any k, we have*

$$\mathbb{E}[f(\boldsymbol{y}_k) - f(\boldsymbol{w}^*)] \le \frac{8LD_t^2}{k(k+1)}$$

if $\mathbb{E}[\|\tilde{\nabla}_s - \nabla f(\boldsymbol{z}_s)\|^2] \le \frac{L^2D_t^2}{N_t(s+1)^2}$ for all $s \le k$.

Proof of Theorem 2. We prove by induction. The base case $t=0$ holds by the exact same argument as in the proof of Theorem 1. Suppose $\mathbb{E}[f(\boldsymbol{w}_{t-1}) - f(\boldsymbol{w}^*)] \le \frac{LD^2}{2^t}$ and consider iteration t. Below we use another induction to prove $\mathbb{E}[f(\boldsymbol{y}_k) - f(\boldsymbol{w}^*)] \le \frac{8LD_t^2}{k(k+1)}$ for any $1 \le k \le N_t$, which will concludes the proof since for any of the three cases, we have $\mathbb{E}[f(\boldsymbol{w}_t) - f(\boldsymbol{w}^*)] = \mathbb{E}[f(\boldsymbol{y}_{N_t}) - f(\boldsymbol{w}^*)]$ which is at most $\frac{8LD_t^2}{N_t^2} \le \frac{LD^2}{2^{t+1}}$.

We first show that the condition $\mathbb{E}[\|\boldsymbol{y}_0 - \boldsymbol{w}^*\|^2] \le D_t^2$ holds. This is trivial for Cases (a) and (b) when $D_t = D$. For Case (c), by strong convexity and the inductive assumption, we have $\mathbb{E}[\|\boldsymbol{y}_0 - \boldsymbol{w}^*\|^2] \le \frac{2}{\alpha}\mathbb{E}[f(\boldsymbol{y}_0) - f(\boldsymbol{w}^*)] \le \frac{LD^2}{\alpha 2^{t-1}} = D_t^2$.

Next note that Lemma 1 implies that $\mathbb{E}[\|\tilde{\nabla}_s - \nabla f(\boldsymbol{z}_s)\|^2]$ is at most $\frac{6L}{m_{t,s}}(2\mathbb{E}[f(\boldsymbol{z}_s) - f(\boldsymbol{w}^*)] + \mathbb{E}[f(\boldsymbol{y}_0) - f(\boldsymbol{w}^*)])$. So the key is to bound $\mathbb{E}[f(\boldsymbol{z}_s) - f(\boldsymbol{w}^*)]$. With $\boldsymbol{z}_1 = \boldsymbol{y}_0$ one can verify that $\mathbb{E}[\|\tilde{\nabla}_1 - \nabla f(\boldsymbol{z}_1)\|^2]$ is at most $\frac{18L}{m_{t,1}}\mathbb{E}[f(\boldsymbol{y}_0) - f(\boldsymbol{w}^*)] \le \frac{18L^2D^2}{m_{t,1}2^t} \le \frac{L^2D_t^2}{4N_t}$ for all three cases, and thus $\mathbb{E}[f(\boldsymbol{y}_s) - f(\boldsymbol{w}^*)] \le \frac{8LD_t^2}{s(s+1)}$ holds for $s=1$ by Lemma 3. Now suppose it holds for any $s < k$, below we discuss the three cases separately to show that it also holds for $s = k$.

Case (a). By smoothness, the condition $\nabla f(\boldsymbol{w}^*) = 0$, the construction of $\boldsymbol{z}_s$, and Cauchy-Schwarz inequality, we have for any $1 < s \le k$,

$$\begin{aligned}
f(\boldsymbol{z}_s) &\le f(\boldsymbol{y}_{s-1}) + (\nabla f(\boldsymbol{y}_{s-1}) - \nabla f(\boldsymbol{w}^*))^\top(\boldsymbol{z}_s - \boldsymbol{y}_{s-1}) \\
&\quad + \frac{L}{2}\|\boldsymbol{z}_s - \boldsymbol{y}_{s-1}\|^2 \\
&= f(\boldsymbol{y}_{s-1}) + \gamma_s(\nabla f(\boldsymbol{y}_{s-1}) - \nabla f(\boldsymbol{w}^*))^\top(\boldsymbol{x}_{s-1} - \boldsymbol{y}_{s-1}) \\
&\quad + \frac{L\gamma_s^2}{2}\|\boldsymbol{x}_{s-1} - \boldsymbol{y}_{s-1}\|^2 \\
&\le f(\boldsymbol{y}_{s-1}) + \gamma_sD\|\nabla f(\boldsymbol{y}_{s-1}) - \nabla f(\boldsymbol{w}^*)\| + \frac{LD^2\gamma_s^2}{2}.
\end{aligned}$$

Property (1) and the optimality of $\boldsymbol{w}^*$ implies:

$$\begin{aligned}
&\|\nabla f(\boldsymbol{y}_{s-1}) - \nabla f(\boldsymbol{w}^*)\|^2 \\
&\le 2L(f(\boldsymbol{y}_{s-1}) - f(\boldsymbol{w}^*) - \nabla f(\boldsymbol{w}^*)^\top(\boldsymbol{y}_{s-1} - \boldsymbol{w}^*)) \\
&\le 2L(f(\boldsymbol{y}_{s-1}) - f(\boldsymbol{w}^*)).
\end{aligned}$$

So subtracting $f(\boldsymbol{w}^*)$ and taking expectation on both sides, and applying Jensen's inequality and the inductive assumption, we have

$$\begin{aligned}
&\mathbb{E}[f(\boldsymbol{z}_s) - f(\boldsymbol{w}^*)] \\
&\le \mathbb{E}[f(\boldsymbol{y}_{s-1}) - f(\boldsymbol{w}^*)] + \gamma_sD\sqrt{2L\mathbb{E}[f(\boldsymbol{y}_{s-1}) - f(\boldsymbol{w}^*)]} \\
&\quad + \frac{2LD^2}{(s+1)^2} \\
&\le \frac{8LD^2}{(s-1)s} + \frac{8LD^2}{(s+1)\sqrt{(s-1)s}} + \frac{2LD^2}{(s+1)^2} < \frac{55LD^2}{(s+1)^2}.
\end{aligned}$$

On the other hand, we have $\mathbb{E}[f(\boldsymbol{y}_0) - f(\boldsymbol{w}^*)] \le \frac{LD^2}{2^t} \le \frac{16LD^2}{(N_t-1)^2} < \frac{40LD^2}{(N_t+1)^2} \le \frac{40LD^2}{(s+1)^2}$. So $\mathbb{E}[\|\tilde{\nabla}_s - \nabla f(\boldsymbol{z}_s)\|^2$ is at most $\frac{900L^2D^2}{m_{t,s}(s+1)^2}$, and the choice of $m_{t,s}$ ensures that this bound is at most $\frac{L^2D^2}{N_t(s+1)^2}$, satisfying the condition of Lemma 3 and thus completing the induction.

Case (b). With the G-Lipschitz condition we proceed similarly and bound $f(\boldsymbol{z}_s)$ by

$$\begin{aligned}
&f(\boldsymbol{y}_{s-1}) + \nabla f(\boldsymbol{y}_{s-1})^\top(\boldsymbol{z}_s - \boldsymbol{y}_{s-1}) + \frac{L}{2}\|\boldsymbol{z}_s - \boldsymbol{y}_{s-1}\|^2 \\
&= f(\boldsymbol{y}_{s-1}) + \gamma_s\nabla f(\boldsymbol{y}_{s-1})^\top(\boldsymbol{x}_{s-1} - \boldsymbol{y}_{s-1}) + \frac{LD^2\gamma_s^2}{2} \\
&\le f(\boldsymbol{y}_{s-1}) + \gamma_sGD + \frac{LD^2\gamma_s^2}{2}.
\end{aligned}$$

So using bounds derived previously and the choice of $m_{t,s}$, we bound $\mathbb{E}[\|\tilde{\nabla}_s - \nabla f(\boldsymbol{z}_s)\|^2$ as follows:

$$\frac{6L}{m_{t,s}}\left(\frac{16LD^2}{(s-1)s} + \frac{4GD}{s+1} + \frac{4LD^2}{(s+1)^2} + \frac{40LD^2}{(s+1)^2}\right) < \frac{6L}{m_{t,s}}\left(\frac{4GD}{s+1} + \frac{116LD^2}{(s+1)^2}\right) < \frac{L^2D^2}{N_t(s+1)^2},$$

again completing the induction.

Case (c). Using the definition of $\boldsymbol{z}_s$ and $\boldsymbol{y}_s$ and direct calculation, one can remove the dependence of $\boldsymbol{x}_s$ and verify

$$\boldsymbol{y}_{s-1} = \frac{s+1}{2s-1}\boldsymbol{z}_s + \frac{s-2}{2s-1}\boldsymbol{y}_{s-2}$$

for any $s \geq 2$. Now we apply Property (2) with $\lambda = \frac{s+1}{2s-1}$:

$$\begin{aligned}
f(\boldsymbol{y}_{s-1}) &\geq \frac{s+1}{2s-1}f(\boldsymbol{z}_s) + \frac{s-2}{2s-1}f(\boldsymbol{y}_{s-2}) \\
&\quad - \frac{L}{2}\frac{(s+1)(s-2)}{(2s-1)^2}\|\boldsymbol{z}_s - \boldsymbol{y}_{s-2}\|^2 \\
&= f(\boldsymbol{w}^*) + \frac{s+1}{2s-1}(f(\boldsymbol{z}_s) - f(\boldsymbol{w}^*)) + \\
&\quad \frac{s-2}{2s-1}(f(\boldsymbol{y}_{s-2}) - f(\boldsymbol{w}^*)) - \frac{L(s-2)}{2(s+1)}\|\boldsymbol{y}_{s-1} - \boldsymbol{y}_{s-2}\|^2 \\
&\geq f(\boldsymbol{w}^*) + \frac{1}{2}(f(\boldsymbol{z}_s) - f(\boldsymbol{w}^*)) - \frac{L}{2}\|\boldsymbol{y}_{s-1} - \boldsymbol{y}_{s-2}\|^2,
\end{aligned}$$

where the equality is by adding and subtracting $f(\boldsymbol{w}^*)$ and the fact $\boldsymbol{y}_{s-1} - \boldsymbol{y}_{s-2} = \frac{s+1}{2s-1}(\boldsymbol{z}_s - \boldsymbol{y}_{s-2})$, and the last inequality is by $f(\boldsymbol{y}_{s-2}) \geq f(\boldsymbol{w}^*)$ and trivial relaxations.

Rearranging gives $f(\boldsymbol{z}_s) - f(\boldsymbol{w}^*) \leq 2(f(\boldsymbol{y}_{s-1} - f(\boldsymbol{w}^*)) + L\|\boldsymbol{y}_{s-1} - \boldsymbol{y}_{s-2}\|^2$. Applying Cauchy-Schwarz inequality, strong convexity and the fact $\mu \geq 1$, we continue with

$$\begin{aligned}
&f(\boldsymbol{z}_s) - f(\boldsymbol{w}^*) \\
&\leq 2(f(\boldsymbol{y}_{s-1} - f(\boldsymbol{w}^*)) \\
&\qquad + 2L(\|\boldsymbol{y}_{s-1} - \boldsymbol{w}^*\|^2 + \|\boldsymbol{y}_{s-2} - \boldsymbol{w}^*\|^2) \\
&\leq 2(f(\boldsymbol{y}_{s-1} - f(\boldsymbol{w}^*)) \\
&\qquad + 4\mu(f(\boldsymbol{y}_{s-1}) - f(\boldsymbol{w}^*) + f(\boldsymbol{y}_{s-2}) - f(\boldsymbol{w}^*)) \\
&\leq 6\mu(f(\boldsymbol{y}_{s-1} - f(\boldsymbol{w}^*)) + 4\mu(f(\boldsymbol{y}_{s-2}) - f(\boldsymbol{w}^*)),
\end{aligned}$$

For $s \geq 3$, we use the inductive assumption to show $\mathbb{E}[f(\boldsymbol{z}_s) - f(\boldsymbol{w}^*)] \leq \frac{48\mu LD_t^2}{(s-1)s} + \frac{32\mu LD_t^2}{(s-2)(s-1)} \leq \frac{448\mu LD_t^2}{(s+1)^2}$. The case for $s = 2$ can be verified similarly using the bound on $\mathbb{E}[f(\boldsymbol{y}_0) - f(\boldsymbol{w}^*)]$ and $\mathbb{E}[f(\boldsymbol{y}_1) - f(\boldsymbol{w}^*)]$ (base case). Finally we bound the term $\mathbb{E}[f(\boldsymbol{y}_0) - f(\boldsymbol{w}^*)] \leq \frac{LD^2}{2^t} = \frac{LD_t^2}{2\mu} \leq \frac{32LD_t^2}{(N_t+1)^2} \leq \frac{32LD_t^2}{(s+1)^2}$, and conclude that the variance $\mathbb{E}[\|\tilde{\nabla}_s - \nabla f(\boldsymbol{z}_s)\|^2$ is at most $\frac{6L}{m_{t,s}}(\frac{896\mu LD_t^2}{(s+1)^2} + \frac{32LD_t^2}{(s+1)^2}) \leq \frac{L^2D_t^2}{N_t(s+1)^2}$, completing the induction by Lemma 3. □

dataset	#features	#categories	#examples
news20	62,061	20	15,935
rcv1	47,236	53	15,564
aloi	128	1,000	108,000

Table 3: Summary of datasets

5. Experiments

To support our theory, we conduct experiments in the multiclass classification problem mentioned in Sec 2.1. Three datasets are selected from the LIBSVM repository[4] with relatively large number of features, categories and examples, summarized in the Table 3.

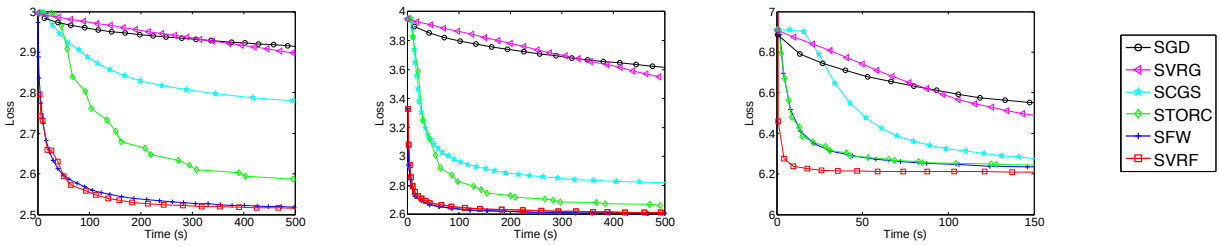
Recall that the loss function is multivariate logistic loss and Ω is the set of matrices with bounded trace norm τ. We focus on how fast the loss decreases instead of the final test error rate so that the tuning of τ is less important, and is fixed to 50 throughout.

We compare six algorithms. Four of them (SFW, SCGS, SVRF, STORC) are projection-free as discussed, and the other two are standard projected stochastic gradient descent (SGD) and its variance-reduced version (SVRG (Johnson & Zhang, 2013)), both of which require expensive projection.

For most of the parameters in these algorithms, we roughly follow what the theory suggests. For example, the size of mini-batch of stochastic gradients at round k is set to k^2, k^3 and k respectively for SFW, SCGS and SVRF, and is fixed to 100 for the other three. The number of iterations between taking two snapshots for variance-reduced methods (SVRG, SVRF and STORC) are fixed to 50. The learning rate is set to the typical decaying sequence $c/\sqrt{k}$ for SGD and a constant c' for SVRG as the original work suggests for some best tuned c and c'.

Since the complexity of computing gradients, performing linear optimization and projecting are very different, we measure the actual running time of the algorithms and see how fast the loss decreases. Results can be found in Figure 1, where one can clearly observe that for all datasets, SGD and SVRG are significantly slower compared to the others, due to the expensive projection step, highlighting the usefulness of projection-free algorithms. Moreover, we also observe large improvement gained from the variance reduction technique, especially when comparing SCGS and STORC, as well as SVF and SVRF on the *aloi* dataset. Interestingly, even though the STORC algorithm gives the best theoretical results, empirically the simpler algorithms

[4] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/



(a) news20 (b) rcv1 (c) aloi

Figure 1: Comparison of six algorithms on three multiclass datasets (best viewed in color)

SFW and SVRF tend to have consistent better performance.

6. Omitted Proofs

Proof of Lemma 1. Let $\mathbb{E}_i$ denotes the conditional expectation given all the past except the realization of i. We have

$$\begin{aligned}
&\mathbb{E}_i[\|\tilde{\nabla} f(\boldsymbol{w};\boldsymbol{w}_0) - \nabla f(\boldsymbol{w})\|^2] \\
&= \mathbb{E}_i[\|\nabla f_i(\boldsymbol{w}) - \nabla f_i(\boldsymbol{w}_0) + \nabla f(\boldsymbol{w}_0) - \nabla f(\boldsymbol{w})\|^2] \\
&= \mathbb{E}_i[\|(\nabla f_i(\boldsymbol{w}) - \nabla f_i(\boldsymbol{w}^*)) - (\nabla f_i(\boldsymbol{w}_0) - \nabla f_i(\boldsymbol{w}^*)) \\
&\quad + (\nabla f(\boldsymbol{w}_0) - \nabla f(\boldsymbol{w}^*)) - (\nabla f(\boldsymbol{w}) - \nabla f(\boldsymbol{w}^*))\|^2] \\
&\le 3\mathbb{E}_i[\|\nabla f_i(\boldsymbol{w}) - \nabla f_i(\boldsymbol{w}^*)\|^2 + \|(\nabla f_i(\boldsymbol{w}_0) - \nabla f_i(\boldsymbol{w}^*)) \\
&\quad - (\nabla f(\boldsymbol{w}_0) - \nabla f(\boldsymbol{w}^*))\|^2 + \|\nabla f(\boldsymbol{w}) - \nabla f(\boldsymbol{w}^*)\|^2] \\
&\le 3\mathbb{E}_i[\|\nabla f_i(\boldsymbol{w}) - \nabla f_i(\boldsymbol{w}^*)\|^2 + \|\nabla f_i(\boldsymbol{w}_0) - \nabla f_i(\boldsymbol{w}^*)\|^2 \\
&\quad + \|\nabla f(\boldsymbol{w}) - \nabla f(\boldsymbol{w}^*)\|^2]
\end{aligned}$$

where the first inequality is Cauchy-Schwarz inequality, and the second one is by the fact $\mathbb{E}_i[\nabla f_i(\boldsymbol{w}_0) - \nabla f_i(\boldsymbol{w}^*)] = \nabla f(\boldsymbol{w}_0) - \nabla f(\boldsymbol{w}^*)$ and that the variance of a random variable is bounded by its second moment.

We now apply Property (1) to bound each of the three terms above. For example, $\mathbb{E}_i\|\nabla f_i(\boldsymbol{w}) - \nabla f_i(\boldsymbol{w}^*)\|^2 \le 2L\mathbb{E}_i[f_i(\boldsymbol{w}) - f_i(\boldsymbol{w}^*) - \nabla f_i(\boldsymbol{w}^*)^\top(\boldsymbol{w} - \boldsymbol{w}^*)] = 2L(f(\boldsymbol{w}) - f(\boldsymbol{w}^*) - \nabla f(\boldsymbol{w}^*)^\top(\boldsymbol{w} - \boldsymbol{w}^*))$, which is at most $2L(f(\boldsymbol{w}) - f(\boldsymbol{w}^*))$ by the optimality of $\boldsymbol{w}^*$. Proceeding similarly for the other two terms concludes the proof. □

Proof of Lemma 2. For any $s \le k$, by smoothness we have $f(\boldsymbol{x}_s) \le f(\boldsymbol{x}_{s-1}) + \nabla f(\boldsymbol{x}_{s-1})^\top(\boldsymbol{x}_s - \boldsymbol{x}_{s-1}) + \frac{L}{2}\|\boldsymbol{x}_s - \boldsymbol{x}_{s-1}\|^2$. Plugging in $\boldsymbol{x}_s = (1-\gamma_s)\boldsymbol{x}_{s-1} + \gamma_s\boldsymbol{v}_s$ gives $f(\boldsymbol{x}_s) \le f(\boldsymbol{x}_{s-1}) + \gamma_s\nabla f(\boldsymbol{x}_{s-1})^\top(\boldsymbol{v}_s - \boldsymbol{x}_{s-1}) + \frac{L\gamma_s^2}{2}\|\boldsymbol{v}_s - \boldsymbol{x}_{s-1}\|^2$. Rewriting and using the fact that $\|\boldsymbol{v}_s - \boldsymbol{x}_{s-1}\| \le D$ leads to

$$\begin{aligned}
f(\boldsymbol{x}_s) &\le f(\boldsymbol{x}_{s-1}) + \gamma_s\tilde{\nabla}_s^\top(\boldsymbol{v}_s - \boldsymbol{x}_{s-1}) \\
&\quad + \gamma_s(\nabla f(\boldsymbol{x}_{s-1}) - \tilde{\nabla}_s)^\top(\boldsymbol{v}_s - \boldsymbol{x}_{s-1}) + \frac{LD^2\gamma_s^2}{2}.
\end{aligned}$$

The optimality of $\boldsymbol{v}_s$ implies $\tilde{\nabla}_s^\top\boldsymbol{v}_s \le \tilde{\nabla}_s^\top\boldsymbol{w}^*$. So with further rewriting we arrive at

$$\begin{aligned}
f(\boldsymbol{x}_s) &\le f(\boldsymbol{x}_{s-1}) + \gamma_s\nabla f(\boldsymbol{x}_{s-1})^\top(\boldsymbol{w}^* - \boldsymbol{x}_{s-1}) \\
&\quad + \gamma_s(\nabla f(\boldsymbol{x}_{s-1}) - \tilde{\nabla}_s)^\top(\boldsymbol{v}_s - \boldsymbol{w}^*) + \frac{LD^2\gamma_s^2}{2}.
\end{aligned}$$

By convexity, term $\nabla f(\boldsymbol{x}_{s-1})^\top(\boldsymbol{w}^* - \boldsymbol{x}_{s-1})$ is bounded by $f(\boldsymbol{w}^*) - f(\boldsymbol{x}_{s-1})$, and by Cauchy-Schwarz inequality, term $(\nabla f(\boldsymbol{x}_{s-1}) - \tilde{\nabla}_s)^\top(\boldsymbol{v}_s - \boldsymbol{w}^*)$ is bounded by $D\|\tilde{\nabla}_s - \nabla f(\boldsymbol{x}_{s-1})\|$, which in expectation is at most $\frac{LD^2}{s+1}$ by the condition on $\mathbb{E}[\|\tilde{\nabla}_s - \nabla f(\boldsymbol{x}_{s-1})\|^2]$ and Jensen's inequality. Therefore we can bound $\mathbb{E}[f(\boldsymbol{x}_s) - f(\boldsymbol{w}^*)]$ by

$$\begin{aligned}
&(1-\gamma_s)\mathbb{E}[f(\boldsymbol{x}_{s-1}) - f(\boldsymbol{w}^*)] + \frac{LD^2\gamma_s}{s+1} + \frac{LD^2\gamma_s^2}{2} \\
&= (1-\gamma_s)\mathbb{E}[f(\boldsymbol{x}_{s-1}) - f(\boldsymbol{w}^*)] + LD^2\gamma_s^2.
\end{aligned}$$

Finally we prove $\mathbb{E}[f(\boldsymbol{x}_k) - f(\boldsymbol{w}^*)] \le \frac{4LD^2}{k+2}$ by induction. The base case is trival: $\mathbb{E}[f(\boldsymbol{x}_1) - f(\boldsymbol{w}^*)]$ is bounded by $(1-\gamma_1)\mathbb{E}[f(\boldsymbol{x}_0) - f(\boldsymbol{w}^*)] + LD^2\gamma_1^2 = LD^2$ since $\gamma_1 = 1$. Suppose $\mathbb{E}[f(\boldsymbol{x}_{s-1}) - f(\boldsymbol{w}^*)] \le \frac{4LD^2}{s+1}$ then with $\gamma_s = \frac{2}{s+1}$ we bound $\mathbb{E}[f(\boldsymbol{x}_s) - f(\boldsymbol{w}^*)]$ by

$$\frac{4LD^2}{s+1}\left(1 - \frac{2}{s+1} + \frac{1}{s+1}\right) \le \frac{4LD^2}{s+2},$$

completing the induction. □

7. Conclusion and Open Problems

We conclude that the variance reduction technique, previously shown to be highly useful for gradient descent variants, can also be very helpful in speeding up projection-free algorithms. The main open question is, in the strongly convex case, whether the number of stochastic gradients for STORC can be improved from $\mathcal{O}(\mu^2 \ln \frac{1}{\epsilon})$ to $\mathcal{O}(\mu \ln \frac{1}{\epsilon})$, which is typical for gradient descent methods, and whether the number of linear optimizations can be improved from $\mathcal{O}(\frac{1}{\epsilon})$ to $\mathcal{O}(\ln \frac{1}{\epsilon})$.

Acknowledgements The authors acknowledge support from the National Science Foundation grant IIS-1523815 and a Google research award.

References

Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. IEEE, 2009.

Dudik, Miro, Harchaoui, Zaid, and Malick, Jérôme. Lifted coordinate descent for learning with trace-norm regularization. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22, pp. 327–336, 2012.

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

Frostig, Roy, Ge, Rong, Kakade, Sham M, and Sidford, Aaron. Competing with the empirical risk minimizer in a single pass. In *Proceedings of the 28th Annual Conference on Learning Theory*, 2015.

Garber, Dan and Hazan, Elad. A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. *arXiv preprint arXiv:1301.4666*, 2013.

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

Hazan, Elad and Kale, Satyen. Projection-free online learning. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Hazan, Elad, Kale, Satyen, and Shalev-Shwartz, Shai. Near-optimal algorithms for online matrix prediction. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, pp. 38.1–38.13, 2012.

Jaggi, Martin. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 427–435, 2013.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 27*, pp. 315–323, 2013.

Lacoste-Julien, Simon and Jaggi, Martin. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems 29*, pp. 496–504, 2015.

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

Mahdavi, Mehrdad, Zhang, Lijun, and Jin, Rong. Mixed optimization for smooth functions. In *Advances in Neural Information Processing Systems*, pp. 674–682, 2013.

Moritz, Philipp, Nishihara, Robert, and Jordan, Michael I. A linearly-convergent stochastic l-bfgs algorithm. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Statistics*, 2016.

Nesterov, YU. E. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pp. 372–376, 1983.

Zhang, Xinhua, Schuurmans, Dale, and Yu, Yao-liang. Accelerated training for matrix-norm regularization: A boosting approach. In *Advances in Neural Information Processing Systems 26*, pp. 2906–2914, 2012.