Revisiting Projection-Free Optimization for Strongly Convex Constraint Sets

Jarrid Rector-Brooks
2260 Hayward St
Ann Arbor, MI, 48104
University of Michigan, Ann Arbor
jrectorb@umich.edu

Jun-Kun Wang
226 Ferst Drive NW
Atlanta, GA, 30332
Georgia Institute of Technology
jimwang@gatech.edu*

Barzan Mozafari
2260 Hayward St
Ann Arbor, MI, 48104
University of Michigan, Ann Arbor
mozafari@umich.edu

Abstract

We revisit the Frank-Wolfe (FW) optimization under strongly convex constraint sets. We provide a faster convergence rate for FW without line search, showing that a previously overlooked variant of FW is indeed faster than the standard variant. With line search, we show that FW can converge to the global optimum, even for smooth functions that are not convex, but are quasi-convex and locally-Lipschitz. We also show that, for the general case of (smooth) non-convex functions, FW with line search converges with high probability to a stationary point at a rate of $O(\frac{1}{t})$, as long as the constraint set is strongly convex—one of the fastest convergence rates in non-convex optimization.

1 Introduction

A popular family of optimization algorithms are so-called gradient descent algorithms: iterative algorithms that are comprised of a gradient descent step at each iteration, followed by a projection step when there is a feasibility constraint. The purpose of the projection is to ensure that the update vector remains within the feasible set.

In many cases, however, the projection step may have no closed-form and thus requires solving another optimization problem itself (e.g., for $l_{1.5}$ norm balls or matroid polytopes (Hazan and others 2016; Hazan and Kale 2012)), the closed-form may exist but involve an expensive computation (e.g., the SVD of the model matrix for Schatten-1, Schatten-2, and Schatten-∞ norm balls (Hazan and others 2016)), or there may simply be no method available for computing the projection in general (e.g., the convex hull of rotation matrices (Hazan, Kale, and Warmuth 2010), which arises as a constraint set in online learning settings (Hazan, Kale, and Warmuth 2010)). In these scenarios, each iteration of the gradient descent may require many "inner" iterations to compute the projection (Jaggi, Sulovsk, and others 2010; Lacoste-Julien and Jaggi 2015; Hazan and Kale 2012). This makes the projection step quite costly, and can account for much of the execution time of each iteration (e.g., see our technical report (Rector-Brooks, Wang, and Mozafari 2018)).

*The work performed while a student at the University of Michigan, Ann Arbor.

Frank-Wolfe (FW) optimization — In this paper, we focus on FW approaches, also known as *projection-free* or *conditional gradient* algorithms (Frank and Wolfe 1956). Unlike gradient descent, these algorithms avoid the projection step altogether by ensuring that the update vector always lies within the feasible set. At each iteration, FW solves a linear program over a constraint set. Since linear programs have closed-form solutions for most constraint sets, each iteration of FW is, in many cases, more cost effective than conducting a gradient descent step and then projecting it back to the constraint set (Jaggi 2013; Hazan and Kale 2012; Hazan and others 2016).

Another main advantage of FW is the sparsity of its solution. Since the solution of a linear program is always a vertex (i.e., extreme point) of the feasible set (when the set itself is convex), each iteration of FW can add, at most, one new vertex to the solution vector. Thus, at iteration t, the solution is a combination of, at most, $t + 1$ vertices of the feasible set, thereby guaranteeing the sparsity of the eventual solution (Clarkson 2010; Jaggi 2013; 2011).

For these reasons, FW optimization has drawn growing interest in recent years, especially in matrix completion, structural SVM, computer vision, sparse PCA, metric learning, and many other settings (Jaggi, Sulovsk, and others 2010; Lacoste-Julien et al. 2013; Osokin et al. 2016; Wang et al. 2016; Chari et al. 2015; Harchaoui et al. 2012; Hazan and Kale 2012; Shalev-Shwartz, Gonen, and Shamir 2011). Unfortunately, while faster in each iteration, standard FW requires many more iterations to converge than gradient descent, and therefore is slower overall. This is because FW's convergence rate is typically $O\left(\frac{1}{t}\right)$ while that of (accelerated) gradient descent is $O\left(\frac{1}{t^2}\right)$, where t is the number of iterations (Jaggi 2013).

We make several contributions (summarized in Table 1):

1. We revisit a non-conventional variant of FW optimization, called Primal Averaging (PA) (Lan 2013), which has been largely neglected in the past, as it was believed to have the same convergence rate as FW without line search, yet incurring extra computations (i.e., matrix averaging step) at each iteration. However, we discover that, when the constraint set is strongly convex, this non-conventional variant enjoys a much faster convergence rate with high probability, $O(\frac{1}{t^2})$ versus $O(\frac{1}{t})$, which more than com-

	Additional Assumptions about the Loss Function	Constraint Set Assumption	Convergence Rate	Requires Line Search (In Each Iteration)
Convex Loss Function				
This Paper	None	Strongly convex	$O\left(\frac{1}{t^2}\right)$ with high probability	No
State-of-the-Art Result(s)				
(Jaggi 2013)	None	Convex	$O\left(\frac{1}{t}\right)$	No
(Garber and Hazan 2015)	Strongly convex	Strongly convex	$O\left(\frac{1}{t^2}\right)$	**Yes**
(Lacoste-Julien and Jaggi 2015)	Strongly convex	Polytope	$O\left(\exp\left(-t\right)\right)$	**Yes**
(Levitin and Polyak 1966; Demyanov and Rubinov 1970; Dunn 1979)	Norm of the gradient is lower bounded	Strongly convex	$O\left(\exp\left(-t\right)\right)$	No
(Beck and Teboulle 2004)	$f(x) = \|Ax - b\|_2^2$	Convex	$O\left(\exp\left(-t\right)\right)$	No
Quasi-Convex Loss Function				
This Paper	Locally-Lipschitz, Norm of the gradient is lower bounded	Strongly convex	$O\left(\min\left(\frac{1}{t^{1/3}}, \frac{1}{t^{1/2}}\right)\right)$	**Yes**
State-of-the-Art Result(s)				
Does not exist	Does not exist	Does not exist	Does not exist	Does not exist
Non-Convex Loss Function				
This Paper	None	Strongly convex	$O\left(\frac{1}{t}\right)$ with high probability	**Yes**
State-of-the-Art Result(s)				
(Lacoste-Julien 2016)	None	Convex	$O\left(\frac{1}{t^{1/2}}\right)$	No

Table 1: Our contributions compared to the state-of-the-art results for projection-free optimization. Here, t is the number of iterations. For non-convex functions, convergence is defined in terms of a stationary point instead of a global minimum. Note that although our bound is probabilistic for convex loss functions, we use no additional assumptions on the loss function and do *not* require line search, which can be a costly operation for big data (see Section 2).

pensates for its slightly more expensive iterations. This surprising result has important ramifications in practice, as many classification, regression, multitask learning, and collaborative filtering tasks rely on norm constraints that are strongly convex, e.g., generalized linear models with l_p norm, squared loss regression with l_p norm, multitask learning with Group Matrix norm, and matrix completion with Schatten norm (Kim and Xing 2010; Garber and Hazan 2015; Hazan and others 2016).

2. While previous work on FW optimization has generally focused on convex functions, we show that FW with line search can converge to the global optimum, even for smooth functions that are not convex, but are quasi-convex and locally-Lipschitz.
3. We also study the general case of (smooth) non-convex functions, showing that FW with line search can converge to a stationary point at a rate of $O(\frac{1}{t})$ with high probability, as long as the constraint set is strongly convex. To the best of our knowledge, we are not aware of such a fast convergence rate in the non-convex optimization literature.[1]
4. Finally, we conduct extensive experiments on various benchmark datasets, empirically validating our theoretical results, and comparing the actual performance of various FW variants in practice.

[1]Without any assumptions, converging to local optima for continuous non-convex functions is NP-hard (Carmon et al. 2017; Agarwal et al. 2016).

2 Related Work

Table 1 compares the state-of-the-art on projection-free optimization to our contributions.

Convex optimization — Garber and Hazan (Garber and Hazan 2015) show that for strongly convex and smooth loss functions, FW with line search achieves a convergence rate of $O(\frac{1}{t^2})$ over strongly convex sets. In contrast, we do not need the loss function to be strongly convex. Further, they require an exact line search at each iteration to achieve this convergence rate. Line search, however, comes with significant downsides. An exact line search solves the problem $\min_{\gamma \in [0,1]} f(x + \gamma v)$ for loss function f, solution vector $x \in \mathbb{R}^n$, and descent direction $v \in \mathbb{R}^n$. There are several methods for solving this optimization, and choosing the best method is often difficult for practitioners (e.g., bracketing line searches versus interpolation ones). Moreover, at best, these methods converge to the minimum at a rate of $O\left(\frac{1}{t^2}\right)$ (Sun and Yuan 2006). Approximate line searches require fewer iterations. However, in using them, one loses most theoretical guarantees provided in previous work, including that of (Garber and Hazan 2015). Nonetheless, both exact and inexact line searches involve at least one evaluation of the loss function or one of its derivatives, which can be quite prohibitive for large datasets (see Section 7.2). This is because the underlying

function for data modeling is typically in the form of a finite sum (e.g., regression loss) over all the data. In comparison, Primal Averaging, which we study and promote, does not require a line search and works with a predefined step size. Notably, this allows PA to considerably outperform FW with line search (see Section 7.2).

Prior work (Levitin and Polyak 1966; Demyanov and Rubinov 1970; Dunn 1979) shows that standard FW without line search for smooth functions can achieve an exponential convergence rate, by making a strict assumption that the gradient is lower-bounded everywhere in the feasible set. In our analysis of PA, however, we do not assume the gradient is lower-bounded everywhere, allowing our result to be more widely applicable.

Quasi-convex optimization — Hazan et al. study quasi-convex and locally-Lipschitz loss functions that admit some saddle points (Hazan, Levy, and Shalev-Shwartz 2015). One of the optimization algorithms for this class of functions is the so-called *normalized gradient descent*, which converges to an ϵ-neighborhood of the global minimum. The analysis in (Hazan, Levy, and Shalev-Shwartz 2015) is for unconstrained optimization. In this paper, we analyze FW for the same class of functions, but with strongly convex constraint sets. Interestingly, when the constraint set is an l_2 ball, FW becomes equivalent to normalized gradient descent. In this paper, we both 1) show that FW can converge to a neighborhood of a global minimum, and 2) derive a convergence rate. (Dunn 1979) extends the analysis of FW to a class of quasi-convex functions of the form $f(w) := g(h(w))$, where h is differentiable and monotonically increasing, and g is a smooth function. Such functions are quite rare in machine learning. In contrast, we study a much more general class of quasi-convex functions, including several popular models (e.g., generalized linear models with a sigmoid loss).

Non-convex optimization — While there has been a surge of research on non-convex optimization in recent years (Carmon et al. 2017; Ge et al. 2015; Agarwal et al. 2016; Lee et al. 2016; Lacoste-Julien 2016), nearly all of it has focused on unconstrained optimization. To our knowledge, there are only a few exceptions (Lacoste-Julien 2016; Ghadimi and Lan 2016; Ge et al. 2015; Reddi et al. 2016). (Lacoste-Julien 2016) proves that FW for smooth non-convex functions converges to a stationary point, at a rate of $O(\frac{1}{\sqrt{t}})$, which matches the rate of projected gradient descent. (Reddi et al. 2016) extends this and considers a stochastic version of FW for smooth non-convex functions. Furthermore, Theorem 7 of (Yu, Zhang, and Schuurmans 2014) provides a convergence rate for non-convex optimization using FW, which is slower than $O(\frac{1}{\sqrt{t}})$. We show in this paper that, for strongly convex sets, FW converges to a stationary point with high probability much faster: $O(\frac{1}{t})$.

3 Background

3.1 Preliminaries

Strongly convex constraint sets are quite common in machine learning. For example, when $p \in (1, 2]$, l_p balls $\{u \in \mathbb{R}^n : \|u\|_p \leq r\}$ and Schatten-p balls $\{X \in \mathbb{R}^{m\times n} : \|X\|_{\mathbb{S}_p} \leq r\}$ are all strongly convex (Garber and Hazan 2015), where $\|X\|_{\mathbb{S}_p} = \left(\sum_{i=1}^{\min(m,n)} \sigma(X)_i^p\right)^{1/p}$ is the Schatten-p norm and $\sigma(X)_i$ is the i^{th} largest singular value of X. Group $l_{p,q}$ balls, used in multitask learning (Garber and Hazan 2015; Kim and Xing 2010), are also strongly convex when $p, q \in (1, 2]$. In this paper, we use the following definitions.

Definition 1 (**Strongly convex set**). *A convex set $\Omega \subseteq \mathbb{R}^d$ is an* α-strongly convex set *with respect to a norm $\|\cdot\|$ if for any $u, v \in \Omega$ and any $\theta \in [0, 1]$, the ball induced by $\|\cdot\|$ which is centered at $\theta u + (1-\theta)v$ with radius $\theta(1-\theta)\frac{\alpha}{2}\|u-v\|^2$ is also included in Ω.*

Definition 2 (**Quasi-convex functions**). *A function $f : \mathbb{R}^d \to \mathbb{R}$ is quasi-convex if for all $u, v \in \mathbb{R}^d$ such that $f(u) \leq f(v)$, it follows that $\langle \nabla f(v), u - v\rangle \leq 0$, where $\langle\cdot,\cdot\rangle$ is the standard inner product.*

Definition 3 (**Strictly-quasi-convex functions**). *A function $f : \mathbb{R}^d \to \mathbb{R}$ is strictly-quasi-convex if it is quasi-convex and its gradients only vanish at the global minimum. That is, for all $u \in \mathbb{R}^d$, it follows that $f(u) > f(u^*) \Rightarrow \|\nabla f(u)\| \neq 0$ where u^* is the global minimum.*

Definition 4 (**Strictly-locally-quasi-convex functions**). *Let $u, v \in \mathbb{R}^d, \kappa, \epsilon > 0$. Further, write $\mathbb{B}_r(x)$ as the Euclidean norm ball centered at x of radius r where $x \in \mathbb{R}^d$ and $r \in \mathbb{R}$. We say $f : \mathbb{R}^d \to \mathbb{R}$ is (ϵ, κ, v)-strictly-locally-quasi-convex in u if at least one of the following applies:*

1. $f(u) - f(v) \leq \epsilon$
2. $\|\nabla f(u)\| > 0$ *and for every* $y \in \mathbb{B}_{\frac{\epsilon}{\kappa}}(v)$ *it holds that* $\langle \nabla f(u), y - u\rangle \leq 0$

3.2 A Brief Overview of Frank-Wolfe (FW)

The Frank-Wolfe (FW) algorithm (Algorithm 1) attempts to solve the constrained optimization problem $\min_{x\in\Omega} f(x)$ for some convex constraint set Ω (a.k.a. feasible set) and some function $f : \Omega \to \mathbb{R}$. FW begins with an initial solution $w_0 \in \Omega$. Then, at each iteration, it computes a search direction v_t by minimizing the linear approximation of f at w_t, $v_t = \min_{v\in\Omega}\langle v, \nabla f(w_t)\rangle$, where $\nabla f(w_t)$ is the gradient of f at w_t. Next, FW produces a convex combination of the current iterate w_t and the search direction v_t to find the next iterate $w_{t+1} = (1-\gamma_t)w_t + \gamma_t v_t$ where $\gamma_t \in [0, 1]$ is the learning rate for the current iteration. There are a number of ways to choose the learning rate γ_t. Chief among these are setting $\gamma_t = \frac{2}{t+1}$ (Algorithm 1, option A) or finding γ_t via line search (Algorithm 1, option B).

4 Faster Convergence Rate for Smooth Convex Functions

4.1 Primal Averaging (PA)

PA (Lan 2013) (Algorithm 2) is a variant of FW that operates in a style similar to Nesterov's acceleration method. PA maintains three sequences, $(z_{t-1})_{t=1,2,\ldots}$, $(v_t)_{t=1,2,\ldots}$, and $(w_t)_{t=1,2,\ldots}$. The first is the accelerating sequence (as in Nesterov acceleration), the second is the sequence of search directions, and the third is the sequence of solution vectors. At each iteration, PA updates its sequences by computing two

Algorithm 1 Standard Frank-Wolfe algorithm

Input: loss $f : \Omega \to \mathbb{R}$.
Input: linear opt. oracle $\mathcal{O}(\cdot)$ for Ω.
Initialize: any $w_1 \in \Omega$.
for $t = 1, 2, 3, \ldots$ **do**
  $v_t \leftarrow \mathcal{O}(\nabla f(w_t)) = \arg\min_{v \in \Omega} \langle v, \nabla f(w_t) \rangle$.
  Option (A): Predefined decay learning rate $\{\gamma_t \in [0, 1]\}_{t=1,2,\ldots}$
  Option (B): $\gamma_t = \arg\min_{\gamma \in [0,1]} \gamma \langle v_t - w_t, \nabla f(w_t) \rangle + \gamma^2 \frac{L}{2} \|v_t - w_t\|^2$.
  $w_{t+1} \leftarrow (1 - \gamma_t) w_t + \gamma_t v_t$.
end for

Algorithm 2 Primal Averaging

Initialize any $v_0 \in \Omega \subset \mathbb{R}^d$. Set $w_0 = v_0$.
for $t = 1, 2, 3, \ldots$ **do**
  $\gamma_t = \frac{2}{t+1}$.
  $z_{t-1} = (1 - \gamma_t) w_{t-1} + \gamma_t v_{t-1}$.
  Option (A): $p_t = \Sigma_{i=1}^t \frac{\theta_i}{\Theta_t} \nabla f(z_{i-1})$, where $\Theta_t = \Sigma_{i=1}^t \theta_i$, $\theta_t = t$, and $\frac{\theta_t}{\Theta_t} = \gamma_t$.
  Option (B): $p_t = \nabla f(z_{t-1})$.
  $v_t = \arg\min_{v \in \Omega} \langle v, p_t \rangle$.
  $w_t = (1 - \gamma_t) w_{t-1} + \gamma_t v_t$.
end for

convex combinations and consulting the linear oracle, such that

$$z_{t-1} = (1 - \gamma_t) w_{t-1} + \gamma_t v_{t-1}$$
$$v_t = \arg\min_{v \in \Omega} \langle \Theta_t^{-1} \sum_{i=1}^t \theta_i \nabla f(z_{i-1}), v \rangle$$
$$w_t = (1 - \gamma_t) w_{t-1} + \gamma_t v_t$$

where $\Theta_t = \sum_{i=1}^t \theta_i$ and the θ_i are chosen, such that $\gamma_t = \frac{\theta_t}{\Theta_t}$. Note that choosing θ_t does not require significant computation as setting $\theta_t = t$ satisfies the requirement $\gamma_t = \frac{\theta_t}{\Theta_t}$ for all t. [2]

Since z_{t-1} and w_t are convex combinations of elements of the constraint set Ω, z_{t-1} and w_t are themselves in Ω. While the input to the linear oracle is a single gradient vector in standard FW, PA uses an average of the gradients seen in iterations $1, 2, \ldots, t$ as the input to the linear oracle.

In standard FW, the sequence $(w_t)_{t=1,2,\ldots}$ has the following property (Jaggi 2013; Lan 2013; Hazan and others 2016):

$$f(w_t) - f(w^*) \leq \frac{2L}{t(t+1)} \Sigma_{i=1}^t \|v_i - w_{i-1}\|^2 \quad (1)$$

where w^* is an optimal point and L is the smoothness parameter of f. We observe that the $\frac{1}{t} \sum_{i=1}^t \|v_i - w_{i-1}\|$ factor of (1) is the average distance between the search direction and solution vector pairs. Denote the diameter D of Ω as $D = \sup_{u,v \in \Omega} \|u - v\|$. Then, since w_{i-1} and v_i are both in Ω, we find that $\frac{1}{t} \sum_{i=1}^t \|v_i - w_{i-1}\| \leq D$. That is, the average distance of v_i and w_{i-1} is upper bounded by diameter D of Ω.

[2] If $\theta_t = t$ then $\frac{\theta_t}{\Theta_t} = \frac{t}{\sum_{i=1}^t i} = \frac{2t}{t(t+1)} = \frac{2}{t+1} = \gamma_t$.

Combining this with (1) yields standard FW's convergence rate:

$$\begin{aligned} f(w_t) - f(w^*) &\leq \frac{2L}{t(t+1)} \Sigma_{i=1}^t \|v_i - w_{i-1}\|^2 \\ &\leq \frac{2LD^2}{t+1} = O\left(\frac{1}{t}\right) \end{aligned} \quad (2)$$

PA has a similar guarantee for the sequence $(w_t)_{t=1,2,\ldots}$ (Lan 2013). Namely

$$f(w_t) - f(w^*) \leq \frac{2L}{t(t+1)} \Sigma_{i=1}^t \|v_i - v_{i-1}\|^2 \quad (3)$$

While the inability to guarantee an arbitrarily small distance between v_i and w_i in Equation 1 caused standard FW to converge as $O(\frac{1}{t})$, this is not the case for the distance between v_i and v_{i-1} in Equation 3. Should we be able to bound the distance $\|v_i - v_{i-1}\|$ to be arbitrarily small, we can show that PA converges as $O(\frac{1}{t^2})$ with high probability. We observe that the sequence $(v_t)_{t=1,2,\ldots}$ expresses this behavior when the constraint set is strongly convex. We have the following theorem.[3]

Theorem 1. *Assume the convex function f is smooth with parameter L. Further, define the function h as $h(w) = f(w) + \theta \xi^T w$ where $\theta \in \left(0, \frac{\epsilon}{4D}\right]$, $\xi \in \mathbb{R}^d$, $w \in \Omega$, Ω is an α-strongly convex set, D is the diameter of Ω, and ξ is uniform on the unit sphere. Applying PA to h yields the following convergence rate for f with probability $1 - \delta$,*

$$f(w_t) - f(w^*) = O\left(\frac{dL}{\alpha^2 \delta^2 t^2}\right)$$

[3] All omitted proofs can be found in our technical report (Rector-Brooks, Wang, and Mozafari 2018).

Theorem 1 states that applying PA to a perturbed function h over an α-strongly convex constraint set allows any smooth, convex function f to converge as $O\left(\frac{1}{t^2}\right)$ with probability $1-\delta$, albeit depending on δ and d. However, as t grows, the t^2 term in the convergence rate's denominator quickly dominates the rate's δ and d terms. This, combined with PA's non-reliance on line search, allows it to outperform the method proposed in (Garber and Hazan 2015). We note that, although Theorem 1 requires us to run PA on the perturbed function h, *f itself still converges as $O\left(\frac{1}{t^2}\right)$ with high probability*. That is, the iterates w_t produced by running PA on h themselves have the guarantee of $f(w_t) - f(w^*) = O\left(\frac{dL}{\alpha^2\delta^2 t^2}\right)$ for $w^* = \arg\min_{w\in\Omega} f(w)$ with probability $1-\delta$. We also empirically investigate this result in Section 7.

4.2 Stochastic Primal Averaging (SPA)

Here we provide a stochastic version of Primal Averaging. While in the previous section we studied PA with Option (A) of Algorithm 2, we now consider PA with Option (B) of Algorithm 2, providing an analysis of its stochastic version. That is, $p_t = \tilde{\nabla} f(z_{t-1})$, where $\tilde{\nabla} f$ represents the aggregated stochastic gradient constructed as $\tilde{\nabla} f(z_{t-1}) = \sum_{i\in S_t} \hat{\nabla} f_i(z_{t-1})$. Further, $\hat{\nabla} f_i(\cdot)$ is the stochastic gradient computed with the ith item of a dataset of size N, while S_t is the set of indices sampled without replacement from $\{1, 2, \ldots, N\}$ at iteration t. We note that $|S_t| = \min(t^4, N)$.

Theorem 2. *Assume the convex function f is smooth with parameter L. Denote σ as the variance of a stochastic gradient. Suppose $p_t = \tilde{\nabla} f(z_{t-1})$ and the number of samples used to obtain p_t is $n_t = O(t^4)$. Further, define the function h as $h(w) = f(w) + \theta\xi^T w$ where $\theta \in \left(0, \frac{\epsilon}{4D}\right]$, $\xi \in \mathbb{R}^d$, $w \in \Omega$, Ω is an α-strongly convex set, D is the diameter of Ω, and ξ is uniform on the unit sphere. Then applying PA to h yields the following convergence rate for f with probability $1-\delta$,*

$$E[f(w_t)] - f(w^*) = O\left(\frac{dL^2(D^2+\sigma)\log t}{\alpha^2\delta^2 t^2}\right)$$

Theorem 2 states that the stochastic version of PA maintains an $O\left(\frac{\log t}{t^2}\right)$ convergence rate with high probability, using h in a manner similar to Theorem 1. Note that n_t grows as $O(t^4)$ until it begins to use all the data points to compute the gradient. Thus, for earlier iterations of SPA, the algorithm requires far less computation than its deterministic counterpart. However, the samples required in each iteration grows quickly, causing later iterations of SPA to share the same computational cost as deterministic Primal Averaging.

5 Strictly-Locally-Quasi-Convex Functions

In this section we show that FW with line search can converge within an ϵ-neighborhood of the global minimum for strictly-locally-quasi-convex functions. Furthermore, if it is assumed that the norm of the gradient is lower bounded, then FW with line search can converge within an ϵ-neighborhood of the global minimum in $O\left(max\left(\frac{1}{\epsilon^2}, \frac{1}{\epsilon^3}\right)\right)$ iterations.

Theorem 3. *Assume that the function f is smooth with parameter L, and that f is (ϵ, κ, w^*)-strictly-locally-quasi-convex, where w^* is a global minimum. Then, the standard FW algorithm with line search (Algorithm 1 option (B)) can converge within an ϵ-neighborhood of the global minimum when the constraint set is strongly convex. Furthermore, if one assumes that $f(w) - f(w^*) \geq \epsilon$ implies that the norm of the gradient is lower bounded as $\|\nabla f(w)\| \geq \theta\epsilon$ for some $\theta \in \mathbb{R}$, then the algorithm needs $t = O(max(\frac{2\kappa}{\theta\epsilon^2}, \frac{8L\kappa}{\theta\epsilon^3}))$ iterations to produce an iterate that is within an $\epsilon-$neighborhood of the global minimum.*

Hazan et al. (Hazan, Levy, and Shalev-Shwartz 2015) provide several examples of strictly-locally-quasi-convex functions. First, if $\epsilon \in (0, 1]$ and $x = (x_1, x_2) \in [-10, 10]^2$, then the function

$$g(x) = (1 + e^{-x_1})^{-1} + (1 + e^{-x_2})^{-1}$$

is $(\epsilon, 1, x^*)$-strictly-locally-quasi-convex in x. Second, if $\epsilon \in (0, 1)$ and $w \in \mathbb{R}^d$, then the function

$$h(w) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \phi(\langle w, x_i\rangle))^2$$

is $(\epsilon, \frac{2}{\gamma}, w^*)$-strictly-locally-quasi-convex in w. Here, $\phi(z) = \mathbb{1}_{z\geq 0}$, $\gamma \in \mathbb{R}$ is the margin of a perceptron, and we have m samples $\{(x_i, y_i)\}_{i=1}^m \in \mathbb{B}_1(0) \times \{0, 1\}$ where $\mathbb{B}_1(0) \subset \mathbb{R}^d$.

6 Smooth Non-Convex Functions

In this section, we show that, with high probability, FW with line search converges as $O\left(\frac{1}{t}\right)$ to a stationary point when the loss function is non-convex and the constraint set is strongly convex. To our knowledge, a rate this rapid does not exist in the non-convex optimization literature.

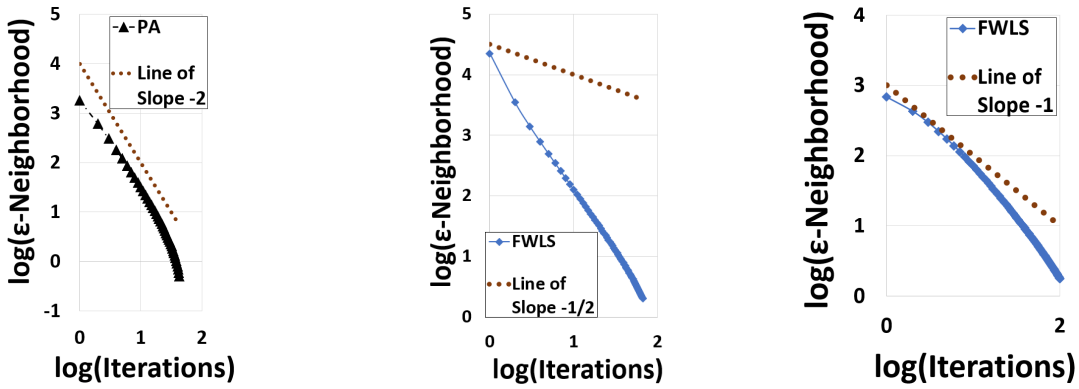
To help demonstrate our theoretical guarantee, we introduce a measure called the FW gap. The FW gap of f at a point $w_t \in \Omega$ is defined as $k_t := \max_{v\in\Omega}\langle v - w_t, -\nabla f(w_t)\rangle$. This measure is adopted in (Lacoste-Julien 2016), which is the first work to show that, for smooth non-convex functions, FW has an $O\left(\frac{1}{\sqrt{t}}\right)$ convergence rate to a stationary point over arbitrary convex sets. The $O\left(\frac{1}{\sqrt{t}}\right)$ rate matches the rate of projected gradient descent when the loss function is smooth and non-convex. It has been shown (Lacoste-Julien 2016) that a point w_t is a stationary point for the constrained optimization problem if and only if $k_t = 0$.

Theorem 4. *Assume that the non-convex function f is smooth with parameter L and the constraint set Ω is α-strongly convex and has dimensionality d. Further, define the function h as $h(w) = f(w) + \theta\xi^T w$ where $\theta \in \left(0, \frac{\epsilon}{4D}\right]$, $\xi \in \mathbb{R}^d$, $w \in \Omega$, D is the diameter of Ω, and ξ is uniform on the unit sphere. Let $\ell_1 = f(w_1) - f(w^*)$ and $C' = \frac{\alpha\delta\sqrt{\pi}}{8L\sqrt{2d}}$. Then applying FW with line search to h yields the following guarantee for the FW gap of f with probability $1-\delta$,*

$$\min_{1\leq s\leq t} k_s \leq \frac{\ell_1}{t\min\{\frac{1}{2}, C'\}} = O\left(\frac{1}{t}\right)$$

Convexity of Loss Function	Loss Function	Constraint	Task
Convex	Quadratic Loss	l_p norm	Regression
	Observed Quadratic Loss	Schatten-p norm	Matrix Completion
Strictly-Locally-Quasi-Convex	Squared Sigmoid	l_p norm	Classification
Non-Convex	Bi-Weight Loss	l_p norm	Robust Regression

Table 2: Various loss functions and constraint sets used in our experiments.



(a) Matrix completion w/ convex (observed quadratic) loss, Schatten-2 norm constraint.

(b) Classification w/ quasi-convex (squared sigmoid) loss, l_2 norm constraint.

(c) Regression w/ non-convex (bi-weight) loss, l_2 norm constraint.

Figure 1: Convergence rates of FW variants for convex loss without line search and non-convex loss with line search.

We would further discuss the result stated in the theorem. In non-convex optimization literature, Nesterov and Polyak (Nesterov and Polyak 2006) show that cubic regularization of Newton's method can find a stationary point in $O(\epsilon^{-3/2})$ iterations and evaluations of the Hessian. First order methods, such as gradient descent, typically require $O(\epsilon^{-2})$ iterations (Carmon et al. 2017) to converge to a stationary point. Recent progress on first order methods, however, assumes some mild conditions and show that an improved rate of $O(\epsilon^{-7/4})$ is possible (Carmon et al. 2017; Agarwal et al. 2016). Here, we show that when the constraint set is strongly convex, FW with line search only needs $O(\epsilon^{-1})$ iterations to arrive within an ϵ-neighborhood of a stationary point. It is important to note, although the $O(\epsilon^{-1})$ convergence rate holds probabilistically, it is quite fast compared to the known rates in the non-convex optimization literature.

7 Experiments

We have conducted extensive experiments on different combinations of loss functions, constraint sets, and real-life datasets (Table 2). Here, we only report two main sets of experiments: the empirical validation of our theoretical results in terms of convergence rates (Section 7.1) and the comparison of various optimizations in terms of actual run times (Section 7.2). We refer the interested reader to our technical report for additional experiments (Rector-Brooks, Wang, and Mozafari 2018).

For classification and regression, we used the logistic and quadratic loss functions. For matrix completion, we used the **observed quadratic loss** (Freund, Grigas, and Mazumder 2017), defined as $f(X) = \sum_{(i,j)\in P(M)}(X_{i,j} - M_{i,j})^2$ where X is the estimated matrix, M is the observed matrix, and $P(M)=\{(i,j) : M_{i,j}$ is observed$\}$. As a non-convex, but strictly-locally-quasi-convex loss, we also used **squared sigmoid loss** $\varphi(z) = (1+\exp(-z))^{-1}$ (Hazan, Levy, and Shalev-Shwartz 2015) for classification. For robust regression, we used the **bi-weight loss** (Belagiannis et al. 2015), as a non-convex (but smooth) loss $\psi(f(x_i), y_i) = \frac{(f(x_i)-y_i)^2}{1+(f(x_i)-y_i)^2}$.

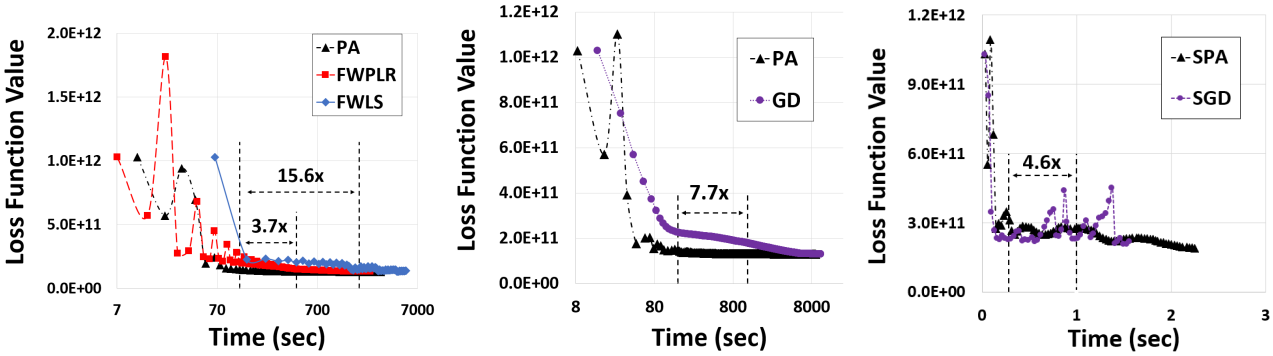
For regression, we used the YearPredictionMSD dataset (500K observations, 90 features) (Lichman 2013). For classification, we used the Adult dataset (49K observations, 14 features) (Lichman 2013). For matrix completion, we used the MovieLens dataset (1M movie ratings from 6,040 users on 3,900 movies) (Harper and Konstan 2016).

7.1 Empirical Validation of Convergence Rates

We ran several experiments to empirically validate our convergence results. In particular, we studied the performance of Primal Averaging (**PA**) and standard FW With Line Search (**FWLS**) with both l_2 and Schatten-2 norm balls as our strongly convex constraint sets.

Theorem 1 guarantees a convergence rate of $O(\frac{1}{t^2})$ for PA when the constraint set is strongly convex and the loss function is convex. We experimented with both l_2 (logistic classifier) and Schatten-2 norm (matrix completion) balls, measuring the loss value at each iteration. As shown in Figure 1a, a slope of -2.41 confirms Theorem 1's guarantee, which predicts a slope of at least -2.

Theorem 3 shows that FWLS converges to the global minimum at the rate of $O\left(\min\left(\frac{1}{t^{1/3}}, \frac{1}{t^{1/2}}\right)\right)$ when the constraint



(a) PA vs. standard FW variants. (b) PA vs. gradient descent. (c) Stochastic PA vs. stochastic GD.

Figure 2: PA versus (a) other FW variants, (b) gradient descent, and (c) stochastic gradient descent.

set is strongly convex and the loss function is strictly-locally-quasi-convex. We investigated this result with the squared sigmoid loss and an l_2 norm constraint. Figure 1b exhibits our results, showing a slope of -2.12, a finding better than the worst-case bounds given by Theorem 3, i.e., a slope of -0.5 (see our technical report (Rector-Brooks, Wang, and Mozafari 2018) for a detailed discussion).

From Theorem 4, we expect FWLS to converge to a stationary point of a (smooth) non-convex function at a rate of $O(\frac{1}{t})$ when constrained to a strongly convex set. Using the bi-weight loss and an l_2 norm constraint, we measured the loss value at each iteration. As shown in Figure 1c, the results confirmed our theoretical results, showing an even steeper slope (-1.46 instead of -1, since Theorem 4 only provides a worst-case upper bound).

7.2 Comparison of Different Optimization Algorithms

To compare the actual performance of various optimization algorithms, we measure the run times, instead of the number of iterations to convergence, in order to account for the time spent in each iteration. In Figure 2, dotted vertical lines mark the convergence points of various algorithms.

First, we compared all three variants of FW: PA, standard FW With Predefined Learning Rate (**FWPLR**) defined in Algorithm 1 with option A, and standard FW With Line Search (**FWLS**) defined in Algorithm 1 with option B. All methods were tested on a regression task (quadratic loss) with an ℓ_2 norm ball constraint.

As shown in Figure 2a, PA converged $3.7\times$ and $15.6\times$ faster than FWPLR and FWLS, respectively. This considerable speedup has significant ramifications in practice. Traditionally, PA has been shied away from, due to its slower iterations, while its convergence rate was believed to be the same as the more efficient variants (Lan 2013). However, as proven in Section 4, PA does converge in fewer iterations.

We also compared the run time of PA versus projected gradient descent (regression task with a quadratic loss). We compared their deterministic versions in Figure 2b, where PA converged significantly faster ($7.7\times$), as expected. For a fair comparison of their stochastic versions, **Stochastic Primal Averaging (SPA)** and **Stochastic Gradient Descent (SGD)**, we considered two cases: an l_2 constraint (which has an efficient projection) and $l_{1.1}$ constraint (which has a costly projection). As expected, for an efficient projection, SGD converged $4.6\times$ faster than SPA (Figure 2c), and when the projection was costly, SPA converged $25.1\times$ faster (see (Rector-Brooks, Wang, and Mozafari 2018) for detailed plots).

8 Conclusion

In this paper, we revisited an important class of optimization techniques, FW methods, and offered new insight into their convergence properties for strongly convex constraint sets, which are quite common in machine learning. Specifically, we discovered that, for convex functions, a non-conventional variant of FW (i.e., Primal Averaging) converges significantly faster than the commonly used variants of FW with high probability. We also showed that PA's $O(\frac{1}{t^2})$ convergence rate more than compensates for its slightly more expensive computational cost at each iteration. We further proved that for strictly-locally-quasi-convex functions, FW can converge to within an ϵ-neighborhood of the global minimum in $O\left(max(\frac{1}{\epsilon^2}, \frac{1}{\epsilon^3})\right)$ iterations. Even for non-convex functions, we proved that FW's convergence rate is better than the previously known results in the literature with high probability. These new convergence rates have significant ramifications for practitioners, due to the widespread applications of strongly convex norm constraints in classification, regression, matrix completion, and collaborative filtering settings. Finally, we conducted extensive experiments on real-world datasets to validate our theoretical results and investigate our improvement over existing methods. In summary, we showed that PA reduces optimization time by 2.8–$15.6\times$ compared to standard FW variants, and by 7.7–$25.1\times$ compared to projected gradient descent. Our plan is to integrate PA in machine learning libraries libraries, including our BlinkML project (Park et al. 2018).

9 Acknowledgments

This work is in part supported by the National Science Foundation (grants 1629397 and 1553169).

References

Agarwal, N.; Allen-Zhu, Z.; Bullins, B.; Hazan, E.; and Ma, T. 2016. Finding approximate local minima for nonconvex optimization in linear time. *arXiv preprint arXiv:1611.01146*.

Beck, A., and Teboulle, M. 2004. A conditional gradient method with linear rate of convergence for solving convex linear systems. *Mathematical Methods of Operations Research* 59(2):235–247.

Belagiannis, V.; Rupprecht, C.; Carneiro, G.; and Navab, N. 2015. Robust optimization for deep regression. In *Proceedings of the IEEE International Conference on Computer Vision*, 2830–2838.

Carmon, Y.; Duchi, J.; Hinder, O.; and Sidford, A. 2017. Accelerated methods for non-convex optimization. *https://arxiv.org/pdf/1611.00756.pdf*.

Chari, V.; Lacoste-Julien, S.; Laptev, I.; and Sivic, J. 2015. On pairwise costs for network flow multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5537–5545.

Clarkson, K. L. 2010. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)* 6(4):63.

Demyanov, V. F., and Rubinov, A. M. 1970. Approximate methods in optimization problems. *Elsevier Publishing Company,*.

Dunn, J. C. 1979. Rates of convergence for conditional gradient algorithms near singular and nonsingular extremals. *SIAM Journal on Control and Optimization*.

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

Freund, R. M.; Grigas, P.; and Mazumder, R. 2017. An extended frank–wolfe method with "in-face" directions, and its application to low-rank matrix completion. *SIAM Journal on Optimization* 27(1):319–346.

Garber, D., and Hazan, E. 2015. Faster rates for the frank-wolfe method over strongly-convex sets. In *International Conference on Machine Learning*, 541–549.

Ge, R.; Huang, F.; Jin, C.; and Yuan, Y. 2015. Escaping from saddle points - online stochastic gradient for tensor decomposition. *CoRR* abs/1503.02101.

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

Harchaoui, Z.; Douze, M.; Paulin, M.; Dudik, M.; and Malick, J. 2012. Large-scale image classification with trace-norm regularization. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 3386–3393. IEEE.

Harper, F. M., and Konstan, J. A. 2016. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5(4):19.

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

Hazan, E., et al. 2016. Introduction to online convex optimization. *Foundations and Trends® in Optimization* 2(3-4):157–325.

Hazan, E.; Kale, S.; and Warmuth, M. K. 2010. Learning rotations with little regret. In *COLT*, 144–154.

Hazan, E.; Levy, K.; and Shalev-Shwartz, S. 2015. Beyond convexity: Stochastic quasi-convex optimization. In *Advances in Neural Information Processing Systems*, 1594–1602.

Jaggi, M.; Sulovsk, M.; et al. 2010. A simple algorithm for nuclear norm regularized problems. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 471–478.

Jaggi, M. 2011. Sparse convex optimization methods for machine learning. Technical report, ETH Zürich.

Jaggi, M. 2013. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML (1)*, 427–435.

Kim, S., and Xing, E. P. 2010. Tree-guided group lasso for multi-task regression with structured sparsity. In *ICML*, 543–550.

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

Lacoste-Julien, S.; Jaggi, M.; Schmidt, M.; and Pletscher, P. 2013. Block-coordinate frank-wolfe optimization for structural svms. *ICML*.

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

Lan, G. 2013. The complexity of large-scale convex programming under a linear optimization oracle. *https://arxiv.org/abs/1309.5550*.

Lee, J. D.; Simchowitz, M.; Jordan, M. I.; and Recht, B. 2016. Gradient descent converges to minimizers. *COLT*.

Levitin, E. S., and Polyak, B. T. 1966. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*.

Lichman, M. 2013. UCI machine learning repository.

Nesterov, Y., and Polyak, B. T. 2006. Cubic regularization of newton method and its global performance. *Mathematical Programming* 108(1):177–205.

Osokin, A.; Alayrac, J.-B.; Lukasewitz, I.; Dokania, P.; and Lacoste-Julien, S. 2016. Minding the gaps for block frank-wolfe optimization of structured svms. In *International Conference on Machine Learning*, 593–602.

Park, Y.; Qing, J.; Shen, X.; and Mozafari, B. 2018. BlinkML: Approximate machine learning with probabilistic guarantees y. *Technical Report* `http://web.eecs.umich.edu/~mozafari/php/data/uploads/blinkml_report.pdf`.

Rector-Brooks, J.; Wang, J.-K.; and Mozafari, B. 2018. Revisiting projection-free optimization for strongly convex constraint sets. *Technical Report* `http://web.eecs.umich.edu/~mozafari/php/data/uploads/fw_report.pdf`.

Reddi, S. J.; Sra, S.; Poczos, B.; and Smola., A. 2016. Stochastic frank-wolfe methods for nonconvex optimization. *Allerton*.

Shalev-Shwartz, S.; Gonen, A.; and Shamir, O. 2011. Large-scale convex minimization with a low-rank constraint. *arXiv preprint arXiv:1106.1622*.

Sun, W., and Yuan, Y.-X. 2006. *Optimization theory and methods: nonlinear programming*, volume 1. Springer Science & Business Media.

Wang, Y.-X.; Sadhanala, V.; Dai, W.; Neiswanger, W.; Sra, S.; and Xing, E. 2016. Parallel and distributed block-coordinate frank-wolfe algorithms. In *International Conference on Machine Learning*, 1548–1557.

Yu, Y.; Zhang, X.; and Schuurmans, D. 2014. Generalized conditional gradient for structured estimation. *arXiv:1410.4828*.