Linear Convergence of Stochastic Frank Wolfe Variants

Donald Goldfarb
Columbia University

Garud Iyengar
Columbia University

Chaoxu Zhou
Columbia University

Abstract

In this paper, we show that the Away-step Stochastic Frank-Wolfe (ASFW) and Pairwise Stochastic Frank-Wolfe (PSFW) algorithms converge linearly in expectation. We also show that if an algorithm convergences linearly in expectation then it converges linearly almost surely. In order to prove these results, we develop a novel proof technique based on concepts of empirical processes and concentration inequalities. As far as we know, this technique has not been used previously to derive the convergence rates of stochastic optimization algorithms. In large-scale numerical experiments, ASFW and PSFW perform as well as or better than their stochastic competitors in actual CPU time.

1 INTRODUCTION

1.1 Motivation

The recent trend of using a large number of parameters to model large datasets in machine learning and statistics has created a strong demand for optimization algorithms that have low computational cost per iteration and exploit model structure. Regularized empirical risk minimization (ERM) is an important class of problems in this area that can be formulated as smooth constrained optimization problems. A popular approach for solving such ERM problems is the proximal gradient method which solves a projection sub-problem in each iteration. The major drawback of this method is that the projection step can be expensive in many situations. As an alternative, the Frank-Wolfe (FW) algorithm [Frank and Wolfe, 1956], also known as the conditional gradient method, solves a linear optimization sub-problem in each iteration, which is much faster than the standard projection technique when the feasible set is a simple polytope [Nesterov, 2015]. When the number of observations in ERM is large, calculating the gradient in every FW iteration becomes a computationally intensive task. The question of whether 'cheap' stochastic gradients can be used as a surrogate in FW immediately arises.

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

1.2 Contribution

In this paper, we show that the Away-step Stochastic Frank-Wolfe (ASFW) algorithm converges linearly in expectation and each sample path of the algorithm converges linearly. We also show that if an algorithm converges linearly in expectation then it converges linearly almost surely. To the best of our knowledge, this is the first paper that proves these results. The major technical difficulty of analyzing the ASFW algorithm is the lack of tools that combine stochastic arguments and combinatorial arguments. In order to solve this problem and prove our convergence results, a novel proof technique based on concepts in empirical processes theory and concentration inequalities is developed. This technique is then applied to prove the linear convergence in expectation and almost sure convergence of each sample path of another Frank-Wolfe variant, the Pairwise Stochastic Frank-Wolfe (PSFW) algorithm. We note that this technique may be useful for analyzing the convergence of other stochastic algorithms. In our large-scale numerical experiments, the proposed algorithms outperform their competitors in all different settings.

1.3 Related Work

The Frank-Wolfe algorithm was proposed sixty years ago [Frank and Wolfe, 1956] for minimizing a convex function over a polytope and is known to converge at an $O(1/k)$ rate. In Levitin and Polyak [1966] the same convergence rate was proved for compact convex constraints. When both objective function and the constraint set are strongly convex, Garber and Hazan [2015] proved that the Frank-Wolfe algorithm has an $O(1/k^2)$ rate of convergence with a properly chosen step size. Motivated by removing the influence of "bad" visited vertices, the away-steps variant of the Frank-Wolfe algorithm was proposed in Abadie [1970]. Later, Guelat and Marcotte [1986] showed that this variant converges linearly under the assumption that the objective function is strongly convex and the optimum lies in the interior of the constraint polytope. Recently, Garber

and Hazan [2013] and Lacoste-Julien and Jaggi [2014] extended the linear convergence result by removing the assumption of the location of the optimum and Beck and Shtern [2015] extended it further by relaxing the strongly convex objective function assumption. Stochastic Frank-Wolfe algorithms have been considered by Lan [2013] and Lafond et al. [2015] in which an $O(1/k)$ rate of convergence in expectation is proved. Luo and Hazan [2016] considered the Stochastic Varianced-Reduced Frank-Wolfe method (SVRF) which also has convergence rate $O(1/k)$ in expectation. In addition, the Frank-Wolfe algorithm has been applied to solve several different classes of problems, including non-linear SVM [Ouyang and Gray, 2010], structural SVM [Lacoste-Julien et al., 2013] [Osokin et al., 2016], and comprehensive principal component pursuit [Mu et al., 2015] among many others. To compare FW variants and other useful algorithms such as the Prox-SVRG of Lin and Zhang [2014] and the stochastic variance reduced FW algorithm of Luo and Hazan [2016], we summarize the theoretical performance in Table 1 which includes the required conditions for convergence and the given complexity bounds, the number of exact and stochastic gradient oracle calls, the number of linear optimization oracle (LO) calls and the number of projection calls in order to obtain an ϵ-approximate solution.

1.4 Problem Description

Consider the minimization problem

$$\min_{\mathbf{x}\in\mathcal{P}} \Big\{ F(\mathbf{x}) \equiv \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \Big\}, \tag{P1}$$

where $\mathcal{P}$ is a polytope, i.e., a non-empty compact polyhedron given by $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^p : \mathbf{C}\mathbf{x} \leq \mathbf{d}\}$ for some $\mathbf{C} \in \mathbb{R}^{m\times p}$, $\mathbf{d} \in \mathbb{R}^m$. Therefore, the set of vertices V of the polytope $\mathcal{P}$ has finitely many elements. Let $D = \sup\{\|\mathbf{x}-\mathbf{y}\| \mid \mathbf{x},\mathbf{y} \in \mathcal{P}\}$ be the diameter of $\mathcal{P}$. For every $i = 1,\dots,n$, $f_i : \mathbb{R} \to \mathbb{R}$ is a strongly convex function with parameter σ_i with an L_i Lipschitz continuous gradient. From another point of view, P1 can be reformulated as a stochastic optimization problem as below

$$\min_{\mathbf{x}\in\mathcal{P}} \Big\{ \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \equiv \mathbb{E} f(\xi,\mathbf{x}) \Big\} \tag{SP1}$$

where ξ is a random variable that follows a discrete uniform distribution on $\{1,\dots,n\}$, $f(i,\mathbf{x}) = f_i(\mathbf{x})$ for every $i = 1,\dots,n$ and $\mathbf{x} \in \mathcal{P}$. Furthermore, define $\nabla f(\xi,\mathbf{x}) = \nabla f_\xi(\mathbf{x})$.

1.5 The Frank-Wolfe Algorithm And Its Variants

In contrast to the projected gradient algorithm, the Frank-Wolfe algorithm (also known as conditional gradient algorithm) calls a linear optimization oracle instead of a projection oracle in every iteration. The Frank-Wolfe Algorithm 1 The Frank-Wolfe Algorithm

Input: $\mathbf{x}^{(1)} \in \mathcal{P}$, $F(\cdot)$
for $k = 1, 2, \dots$ **do**
  Set $\mathbf{p}^{(k)} = \arg\min_{\mathbf{s}\in\mathcal{P}} \langle \nabla F(\mathbf{x}^{(k)}), \mathbf{s}\rangle$.
  Set $\mathbf{d}^{(k)} = \mathbf{p}^{(k)} - \mathbf{x}^{(k)}$.
  Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \gamma^{(k)}\mathbf{d}^{(k)}$, where $\gamma^{(k)} = \frac{2}{k+2}$ or obtain by line-search.
end for
Return: $\mathbf{x}^{(k+1)}$.

rithm has become popular recently because it performs a sparse update at each step. For a good review of what was known about the FW algorithm until a few years ago, see Jaggi [2013]. It is well-known that this algorithm converges sub-linearly with rate $O(1/k)$ because of the so-called zig-zagging phenomenon [Lacoste-Julien and Jaggi, 2015]. Especially when the optimal solution $\mathbf{x}^*$ does not lie in the relative interior of $\mathcal{P}$, the FW algorithm tends to zig-zag amongst the vertices that define the facet containing $\mathbf{x}^*$. One way to overcome this zig-zagging problem is to keep tracking of the "active" vertices (the vertices discovered previously in the FW algorithm) and move away from the "worst" of these in some iterations.

The Away-step Frank-Wolfe algorithm (AFW) and the Pairwise Frank-Wolfe algorithm (PFW) are two notable variants based on this idea. After computing the vertex $\mathbf{p}^{(k)} = \arg\min_{\mathbf{x}\in\mathcal{P}}\langle \nabla F(\mathbf{x}^{(k)}), \mathbf{x}\rangle$ by the linear optimization oracle and the vertex $\mathbf{u}^{(k)} = \arg\max_{\mathbf{x}\in U^{(k)}}\langle \nabla F(\mathbf{x}^{(k)}), \mathbf{x}\rangle$ where $U^{(k)}$ is the set of active vertices at iteration k, the AFW algorithm moves away from the one that maximizes the potential increase in $F(\mathbf{x})$ i.e. the increase in the linearized function, while the PFW algorithm tries to take advantages of both vertices and moves in the direction $\mathbf{p}^{(k)} - \mathbf{u}^{(k)}$. Details of the algorithms can be found in Lacoste-Julien and Jaggi [2015].

2 VARIANTS OF STOCHASTIC FRANK-WOLFE ALGORITHM

When the exact gradients is expensive to compute and an unbiased stochastic gradient is easy to obtain, it may be advantageous to use a stochastic gradient in AFW and PFW. We describe the Away-step Stochastic Frank-Wolfe Algorithm (ASFW) and the Pairwise Stochastic Frank-Wolfe Algorithm(PSFW) below. The following algorithm updates a vertex representation of the current iterate and is called in Algorithms 2 and 3.

Table 1: Summary of requirements and performance of algorithms for obtaining an ϵ-approximate solution

Algorithm	Extra conditions	Exact gradients	Stochastic Gradients	LO	Projection
FW	bounded constraint	$O(1/\epsilon)$	NA	$O(1/\epsilon)$	NA
Away-step FW	polytope constraint strongly convex objective	$O(\log(1/\epsilon))$	NA	$O(\log(1/\epsilon))$	NA
Pairwise FW	polytope constraint strongly convex objective	$O(\log(1/\epsilon))$	NA	$O(\log(1/\epsilon))$	NA
SVRF	bounded constraint	$O(\log(1/\epsilon))$	$O(1/\epsilon^2)$	$O(1/\epsilon)$	NA
Prox-SVRG	strongly convex objective	$\log(1/\epsilon)$	$O(m\log(1/\epsilon))$	NA	$O(m\log(1/\epsilon))$
ASFW	polytope constraint strongly convex objective	NA	$O(1/\epsilon^{4\eta})$, $0<\eta<1$	$O(\log 1/\epsilon)$	NA
PSFW	polytope constraint strongly convex objective	NA	$O(1/\epsilon^{(6\|V\|!+2)\zeta})$, $0<\zeta<1$	$O(\log 1/\epsilon)$	NA

Comparisons of algorithms in terms of their requirements and the theoretical performances to get an ϵ-approximate solution. LO denotes for linear optimizations and. In Prox-SVRG, m is the number of iterations in each epoch. In PSFW, $|V|$ is the number of vertices of the polytope constraint.

Algorithm 2 Away-step Stochastic Frank-Wolfe algorithm

1: **Input:** $\mathbf{x}^{(1)} \in V$, f_i and L_i
2: Set $\mu^{(1)}_{\mathbf{x}^{(1)}} = 1$, $\mu^{(1)}_{\mathbf{v}} = 0$ for any $\mathbf{v} \in V/\{\mathbf{x}^{(1)}\}$ and $U^{(1)} = \{\mathbf{x}^{(1)}\}$.
3: **for** $k = 1, 2, \ldots$ **do**
4: Sample $\xi_1, \ldots, \xi_{m^{(k)}} \overset{\text{i.i.d.}}{\sim} \xi$ and set $\mathbf{g}^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} \nabla_{\mathbf{x}} f(\xi_i, \mathbf{x}^{(k)})$, $L^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} L_{\xi_i}$.
5: Compute $\mathbf{p}^{(k)} \in \arg\min_{\mathbf{x}\in\mathcal{P}} \langle \mathbf{g}^{(k)}, \mathbf{x}\rangle$.
6: Compute $\mathbf{u}^{(k)} \in \operatorname{argmax}_{\mathbf{v}\in U^{(k)}} \langle \mathbf{g}^{(k)}, \mathbf{v}\rangle$.
7: **if** $\langle \mathbf{g}^{(k)}, \mathbf{p}^{(k)} + \mathbf{u}^{(k)} - 2\mathbf{x}^{(k)}\rangle \le 0$ **then**
8: Set $\mathbf{d}^{(k)} = \mathbf{p}^{(k)} - \mathbf{x}^{(k)}$ and $\gamma^{(k)}_{\max} = 1$.
9: **else**
10: Set $\mathbf{d}^{(k)} = \mathbf{x}^{(k)} - \mathbf{u}^{(k)}$ and $\gamma^{(k)}_{\max} = \frac{\mu^{(k)}_{\mathbf{u}^{(k)}}}{1-\mu^{(k)}_{\mathbf{u}^{(k)}}}$.
11: **end if**
12: Set $\gamma^{(k)} = \min\{-\frac{\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle}{L^{(k)}\|\mathbf{d}^{(k)}\|^2}, \gamma^{(k)}_{\max}\}$ or determine it by line-search.
13: Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \gamma^{(k)}\mathbf{d}^{(k)}$.
14: Update $U^{(k+1)}$ and $\mu^{(k+1)}$ by Procedure VRU.
15: **end for**
16: **Return:** $\mathbf{x}^{(k+1)}$.

Algorithm 3 Pairwise Stochastic Frank-Wolfe algorithm

1: Replace line 7 to 11 in Algorithm 2 by: $\mathbf{d}^{(k)} = \mathbf{p}^{(k)} - \mathbf{u}^{(k)}$ and $\gamma^{(k)}_{\max} = \mu^{(k)}_{\mathbf{u}^{(k)}}$.

Algorithm 4 Procedure Vertex Representation Update (VRU)

1: **Input:** $\mathbf{x}^{(k)}$, $(U^{(k)}, \boldsymbol{\mu}^{(k)})$, $\mathbf{d}^{(k)}$, $\gamma^{(k)}$, $\mathbf{p}^{(k)}$ and $\mathbf{v}^{(k)}$.
2: **if** $\mathbf{d}^{(k)} = \mathbf{x}^{(k)} - \mathbf{u}^{(k)}$ **then**
3: Update $\mu^{(k)}_{\mathbf{v}} = \mu^{(k)}_{\mathbf{v}}(1+\gamma^{(k)})$ for any $\mathbf{v} \in U^{(k)}/\{\mathbf{u}^{(k)}\}$.
4: Update $\mu^{(k+1)}_{\mathbf{u}^{(k)}} = \mu^{(k)}_{\mathbf{u}^{(k)}}(1+\gamma^{(k)}) - \gamma^{(k)}$.
5: **if** $\mu^{(k+1)}_{\mathbf{u}^{(k)}} = 0$ **then**
6: Update $U^{(k+1)} = U^{(k)}/\{\mathbf{u}^{(k)}\}$
7: **else**
8: Update $U^{(k+1)} = U^{(k)}$
9: **end if**
10: **end if**
11: Update $\mu^{(k+1)}_{\mathbf{v}} = \mu^{(k)}_{\mathbf{v}}(1-\gamma^{(k)})$ for any $\mathbf{v} \in U^{(k)}/\{\mathbf{p}^{(k)}\}$.
12: Update $\mu^{(k+1)}_{\mathbf{p}^{(k)}} = \mu^{(k)}_{\mathbf{p}^{(k)}}(1-\gamma^{(k)}) + \gamma^{(k)}$.
13: **if** $\mu^{(k+1)}_{\mathbf{p}^{(k)}} = 1$ **then**
14: Update $U^{(k+1)} = \{\mathbf{p}^{(k)}\}$.
15: **else**
16: Update $U^{(k+1)} = U^{(k)} \cup \{\mathbf{p}^{(k)}\}$.
17: **end if**
18: (Optional) Carathéodory's theorem can be applied for the vertex representation of $\mathbf{x}^{(k+1)}$ so that $|U^{(k+1)}| = p+1$ and $\boldsymbol{\mu}^{(k+1)} \in \mathbb{R}^{p+1}$.
19: **Return:** $(U^{(k+1)}, \boldsymbol{\mu}^{(k+1)})$

3 CONVERGENCE PROOF

In this section, we will first introduce some lemmas and notation and then state the main theorems in this paper. Full proofs of the lemmas, theorems and corollaries can be found in the supplementary material. Note that, at the

k-th iteration of the algorithms, $m^{(k)}$ i.i.d. samples of ξ are obtained. Define $F^{(k)}(\mathbf{x}) = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} f_{\xi_i}(\mathbf{x})$. It is easy to see that $F^{(k)}$ is Lipschitz continuous with Lipschitz constant $L^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} L_{\xi_i}$ and strongly convex with constant $\sigma^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} \sigma_{\xi_i}$. The following ancillary problem is used in our analysis.

$$\min_{\mathbf{x}\in\mathcal{P}} F^{(k)}(\mathbf{x}), \tag{H1}$$

Let $\mathbf{x}_*^{(k)}$ denote the optimal solution of problem H1, i.e., $\mathbf{x}_*^{(k)} = \operatorname{argmin}_{\mathbf{x}\in\mathcal{P}} F^{(k)}(\mathbf{x})$. The lemma below plays an important role in our proof. We refer to Beck and Shtern [2015] for a detailed proof of this lemma.

Lemma 1. *For any $\mathbf{x} \in \mathcal{P}/\{\mathbf{x}_*^{(k)}\}$ that can be represented as $\mathbf{x} = \sum_{\mathbf{v}\in U^{(k)}} \mu_{\mathbf{v}}\mathbf{v}$ for some $U^{(k)} \subset V$ where $\sum_{\mathbf{v}\in U^{(k)}} \mu_{\mathbf{v}} = 1$ and $\mu_{\mathbf{v}} > 0$ for every $\mathbf{v} \in U^{(k)}$, it holds that,*

$$\max_{\mathbf{u}\in U, \mathbf{p}\in V} \langle \nabla F^{(k)}(\mathbf{x}), \mathbf{u} - \mathbf{p}\rangle \geq \frac{\Omega_{\mathcal{P}}}{|U|} \frac{\langle \nabla F^{(k)}(\mathbf{x}), \mathbf{x} - \mathbf{x}_*^{(k)}\rangle}{\|\mathbf{x} - \mathbf{x}_*^{(k)}\|}.$$

where $|U^{(k)}|$ denotes the cardinality of $U^{(k)}$, V is the set of extreme points of $\mathcal{P}$ and

$$\Omega_{\mathcal{P}} = \frac{\zeta}{\phi}$$

for

$$\zeta = \min_{\mathbf{v}\in V, i\in\{1,\ldots,m\}: a_i > \mathbf{C}_i\mathbf{v}} (d_i - \mathbf{C}_i\mathbf{v}),$$
$$\phi = \max_{i\in\{1,\ldots,m\}/I(V)} \|\mathbf{C}_i\|.$$

Next, we introduce some definitions and lemmas that are common in the empirical processes literature but rarely seen in the optimization literature.

Definition [Bracketing Number] Let $\mathcal{F}$ be a class of functions. Given two functions l and u, the bracket $[l, u]$ is the set of all function f with $l \leq f \leq u$. An ϵ-bracket in L_1 is a bracket $[l, u]$ with $\mathbb{E}|u - l| < \epsilon$. The bracketing number $N_{[]}(\epsilon, \mathcal{F}, L_1)$ is the minimum number of ϵ-brackets needed to cover $\mathcal{F}$. (The bracketing functions l and u must have finite L_1-norms but need not belong to $\mathcal{F}$).

The bracketing number measures the complexity of a function class. The lemma below provides an upper bound for a function class indexed by a finite dimensional bounded set. This result can be found in any empirical processes textbook such as van der Vaart and Wellner [1996]. See the supplementary material for a proof.

Lemma 2. *Let $\mathcal{F} = \{f_\theta \mid \theta \in \Theta\}$ be a collection of measurable functions indexed by a bounded subset $\Theta \subset \mathbb{R}^p$. Denote $D_\Theta = \sup\{\|\theta_1 - \theta_2\| \mid \theta_1, \theta_2 \in \Theta\}$. Suppose that there exists a measurable function g such that*

$$|f_{\theta_1}(\xi) - f_{\theta_2}(\xi)| \leq g(\xi)\|\theta_1 - \theta_2\| \tag{1}$$

for every $\theta_1, \theta_2 \in \Theta$. If $\|g(\xi)\|_1 \equiv \int |g(\xi)| dP < \infty$, then the bracketing numbers satisfy

$$N_{[]}(\epsilon\|g\|_1, \mathcal{F}, L_1) \leq (\frac{\sqrt{p}D_\Theta}{\epsilon})^p$$

for every $0 < \epsilon < D_\Theta$.

Remark: The bracketing number has a very close relationship with the covering number, which is a better known quantity in machine learning. Let $N(\epsilon, \mathcal{F}, L_1)$ be the covering number of the set $\mathcal{F}$; that is, the minimal number of balls of L_1-radius ϵ needs to cover the set $\mathcal{F}$. Then the relation, $N(\epsilon, \mathcal{F}, L_1) \leq N_{[]}(2\epsilon, \mathcal{F}, L_1)$, between covering number and bracketing number always holds. Moreover, this concept is also closely related to the VC-dimension. Usually, constructing and counting the number of brackets for a class of functions is easier to do than computing the minimum number of balls that covers the class.

Based on the bounds on the bracketing number for a function class with bounded index set, we can provide a concentration bound for $\sup_{\mathbf{x}\in\mathcal{P}} |F^{(k)}(\mathbf{x}) - F(\mathbf{x})|$.

Lemma 3. *For any $\delta > 0$ and $0 < \epsilon < \min\{D, \delta/(2L_F)\}$ we have*

$$\mathbb{P}\{\sup_{\mathbf{x}\in\mathcal{P}} |F^{(k)}(\mathbf{x}) - F(\mathbf{x})| \geq \delta\}$$
$$\leq 2K_{\mathcal{P}}(\frac{D}{\epsilon})^p \exp\{-\frac{m^{(k)}(\delta - 2L_F\epsilon)^2}{2(u_F - l_F)^2}\},$$

where $L_F \equiv \min\{L_1, \ldots L_n\}$, $K_{\mathcal{P}} = (\sqrt{p})^p$, $u_F = \max\{\sup_{\mathbf{x}\in\mathcal{P}} f_i(\mathbf{x}) \mid i = 1, \ldots, n\}$ and $l_F = \min\{\inf_{\mathbf{x}\in\mathcal{P}} f_i(\mathbf{x}) \mid i = 1, \ldots, n\}$.

Corollary 1. *When $m^{(k)} \geq 3$,*

$$\mathbb{E}\sup_{\mathbf{x}\in\mathcal{P}} |F^{(k)}(\mathbf{x}) - F(\mathbf{x})| \leq C_1\sqrt{\frac{\log m^{(k)}}{m^{(k)}}}$$

and

$$\mathbb{E}|F^{(k)}(\mathbf{x}_*^{(k)}) - F(\mathbf{x}^*)| \leq C_1\sqrt{\frac{\log m^{(k)}}{m^{(k)}}}$$

where

$$C_1 = 4(|u_F| + |l_F|)K_{\mathcal{P}}D^p \exp\{-p(\log\frac{u_F - l_F}{2\sqrt{2}L_F})\}$$
$$+ (u_F - l_F)\sqrt{p+1}.$$

Lemma 4. *Let $c_i \geq 0$ and $b_i \in \{0, 1\}$ for $i = 1, \ldots, n$. Assume that $\sum_{j=1}^n b_j = m < n$. Then for $0 < a < 1$ we have*

$$\sum_{k=1}^n a^{\sum_{j=k}^n b_j} c_k \leq \sum_{k=1}^m a^{m-k+1} c_k + \sum_{k=m+1}^n c_k. \tag{2}$$

With the developments of the above lemmas and corollary we are ready to state and prove the main results.

Theorem 1. *Let $\{\mathbf{x}^{(k)}\}_{k\geq 1}$ be the sequence generated by Algorithm 2 for solving Problem $(P1)$, N be the number of vertices used to represent $\mathbf{x}^{(k)}$ (if VRU is implemented by using Carathéodory's theorem, $N = p+1$, otherwise $N = |V|$) and F^* be the optimal value of the problem. Let $\rho = \min\{\frac{1}{2}, \frac{\Omega_{\mathcal{P}}^2 \sigma_F}{16N^2 L_F D^2}\}$ where $\sigma_F = \min\{\sigma_1, \ldots, \sigma_n\}$, $L_F = \max\{L_1, \ldots, L_n\}$. Set $m^{(i)} = \lceil 1/(1-\rho)^{2i+2} \rceil$. Then for every $k \geq 1$*

$$\mathbb{E}\{F(\mathbf{x}^{(k+1)}) - F^*\} \leq C_2(1-\beta)^{(k-1)/2}, \qquad (3)$$

where C_2 is a deterministic constant and $0 < \beta < \rho \leq 1/2$.

Proof. At iteration k, let $\mathbf{x}^{(k)}$ denote the current solution, $\xi_1, \ldots, \xi_{m^{(k)}}$ denote the samples obtained in the algorithm, $\mathbf{d}^{(k)}$ denote the direction that the algorithm will take at this step and $\gamma^{(k)}$ denote the step length. Define $F^{(k)}(\mathbf{x}) = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} f(\xi_i, \mathbf{x})$, $\mathbf{x}_*^{(k)} = \arg\min_{\mathbf{x}\in\mathcal{P}} F^{(k)}(\mathbf{x})$ and $F_*^{(k)} = F^{(k)}(\mathbf{x}_*^{(k)})$. Note that $F^{(k)}$ is Lipschitz continuous with Lipschitz constant $L^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} L_{\xi_i}$ and strongly convex with constant $\sigma^{(k)} = \frac{1}{m^{(k)}} \sum_{i=1}^{m^{(k)}} \sigma_{\xi_i}$. In addition, the stochastic gradient $\mathbf{g}^{(k)} = \nabla F^{(k)}(\mathbf{x})$. Using the arguments in Beck and Shtern [2015], we can separate our analysis into the following four cases at iteration k

$(A^{(k)})$ $\quad \gamma_{\max}^{(k)} \geq 1$ and $\gamma^{(k)} \leq 1$.

$(B^{(k)})$ $\quad \gamma_{\max}^{(k)} \geq 1$ and $\gamma^{(k)} \geq 1$.

$(C^{(k)})$ $\quad \gamma_{\max}^{(k)} < 1$ and $\gamma^{(k)} < \gamma_{\max}^{(k)}$.

$(D^{(k)})$ $\quad \gamma_{\max}^{(k)} < 1$ and $\gamma^{(k)} = \gamma_{\max}^{(k)}$.

Let $\delta_{A^{(k)}}$, $\delta_{B^{(k)}}$, $\delta_{C^{(k)}}$ and $\delta_{D^{(k)}}$ be the indicator functions, Beck and Shtern [2015] gives

$$\begin{aligned}
&\delta_{A^{(k)}}\{F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)}\} \\
&\leq \delta_{A^{(k)}}\{(1 - \frac{\Omega_{\mathcal{P}}^2 \sigma_F}{16N^2 L_F D^2})(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)})\} \\
&\delta_{B^{(k)}}\{F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)}\} \\
&\leq \delta_{B^{(k)}}\{\frac{1}{2}(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)})\} \\
&\delta_{C^{(k)}}\{F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)}\} \\
&\leq \delta_{C^{(k)}}\{(1 - \frac{\Omega_{\mathcal{P}}^2 \sigma_F}{16N^2 L_F D^2})(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)})\} \\
&\delta_{D^{(k)}}\{(F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)})\} \\
&\leq \delta_{D^{(k)}}\{F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)}\}.
\end{aligned}$$

Details of the proof of above inequalities can be found in the supplementary material.

Define $\rho = \min\{\frac{1}{2}, \frac{\Omega_{\mathcal{P}}^2 \sigma_F}{16N^2 L_F D^2}\}$. Note that ρ is a deterministic constant between 0 and 1. Therefore we have

$$\begin{aligned}
&F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} \\
&\leq (\{1-\rho)^{\{1-\delta_{D^{(k)}}\}}(F^{(k)}(\mathbf{x}^{(k)}) - F_*^{(k)}) \\
&= (1-\rho)^{\{1-\delta_{D^{(k)}}\}}(F^{(k-1)}(\mathbf{x}^{(k)}) - F_*^{(k-1)}) \\
&+ (1-\rho)^{\{1-\delta_{D^{(k)}}\}}\{F^{(k)}(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k)}) + F(\mathbf{x}^{(k)}) \\
&- F^{(k-1)}(\mathbf{x}^{(k)}) + F^* - F_*^{(k)} + F_*^{(k-1)} - F^*\} \\
&\leq (1-\rho)^{\{1-\delta_{D^{(k)}}\}}(F^{(k-1)}(\mathbf{x}^{(k)}) - F_*^{(k-1)}) + \\
&(1-\rho)^{\{1-\delta_{D^{(k)}}\}}\{|F^{(k)}(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k)})| + |F_*^{(k)} - F^*| \\
&+ |F^{(k-1)}(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k)})| + |F_*^{(k-1)} - F^*|\} \\
&\leq (1-\rho)^{\sum_{i=1}^{k}\{1-\delta_{D^{(i)}}\}}(F^{(0)}(\mathbf{x}^{(1)}) - F_*^{(0)}) + \\
&\sum_{i=1}^{k}(1-\rho)^{\sum_{j=i}^{k}\{1-\delta_{D^{(j)}}\}}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} \\
&- F^*| + |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i-1)} - F^*|\}.
\end{aligned}$$

At iteration k, there are at most $(k+1)/2$ drop steps, i.e., at most $(k+1)/2$ $\delta_{D^{(i)}}$'s equal to 1. Then by Lemma 4, we have

$$\begin{aligned}
&\sum_{i=1}^{k}(1-\rho)^{\sum_{j=i}^{k}\{1-\delta_{D^{(j)}}\}}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F^* \\
&- F_*^{(i)}| + |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i-1)} - F^*|\} \\
&\leq \sum_{i=k/2}^{k}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F^{(i-1)}(\mathbf{x}^{(i)}) \\
&- F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\} \\
&+ \sum_{i=1}^{k/2-1}(1-\rho)^{k/2-i}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} \\
&- F^*| + |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i-1)} - F^*|\}.
\end{aligned}$$

Therefore

$$\begin{aligned}
&F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} \\
&\leq (1-\rho)^{\frac{k-1}{2}}(u_F - l_F) + \sum_{i=k/2}^{k}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| \\
&+ |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\} \\
&+ \sum_{i=1}^{k/2-1}(1-\rho)^{k/2-i}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| \\
&+ |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\}.
\end{aligned}$$

In addition, $F^{(k)}(\mathbf{x}^{(k+1)}) - F_*^{(k)} = F(\mathbf{x}^{(k+1)}) - F^* +$

$(F^{(k)}(\mathbf{x}^{(k+1)}) - F(\mathbf{x}^{(k+1)})) + (F^* - F_*^{(k)})$. Thus

$$
\begin{aligned}
&F(\mathbf{x}^{(k+1)}) - F^* \\
&\leq (1-\rho)^{\frac{k-1}{2}}(u_F - l_F) + \sum_{i=k/2}^{k+1}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| \\
&+ |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\} \\
&+ \sum_{i=1}^{k/2-1}(1-\rho)^{k/2-i}\{|F^{(i)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| \\
&+ |F^{(i-1)}(\mathbf{x}^{(i)}) - F(\mathbf{x}^{(i)})| + |F_*^{(i)} - F^*| + |F_*^{(i-1)} - F^*|\}.
\end{aligned}
$$

Note that for any deterministic $\mathbf{x} \in \mathcal{P}$, we have $\mathbb{E}F^{(k)}(\mathbf{x}) = F(\mathbf{x})$. In addition, by Corollary 1, the following bound holds for every iteration k

$$
\begin{aligned}
&\mathbb{E}|F^{(k)}(\mathbf{x}^{(k)}) - F(\mathbf{x}^{(k)})| \\
&\leq \mathbb{E}\sup_{\mathbf{x}\in\mathcal{P}}|F^{(k)}(\mathbf{x}) - F(\mathbf{x})| \leq C_1\sqrt{\frac{\log m^{(k)}}{m^{(k)}}}
\end{aligned}
$$

and

$$
\mathbb{E}|F_*^{(k)} - F^*| \leq C_1\sqrt{\frac{\log m^{(k)}}{m^{(k)}}}.
$$

Combining all above bounds and use $m^{(i)} = \lceil 1/(1-\rho)^{2i+2}\rceil$, we have

$$
\begin{aligned}
&\mathbb{E}\{F(\mathbf{x}^{(k+1)}) - F^*\} \\
&\leq (1-\rho)^{\frac{k-1}{2}}(u_F - l_F) \\
&\quad + 2C_1\{\sum_{i=k/2}^{k+1}(\sqrt{\frac{\log m^{(i)}}{m^{(i)}}} + \sqrt{\frac{\log m^{(i-1)}}{m^{(i-1)}}}) \\
&\quad + \sum_{i=1}^{k/2-1}(1-\rho)^{k/2-i}(\sqrt{\frac{\log m^{(i)}}{m^{(i)}}} + \sqrt{\frac{\log m^{(i-1)}}{m^{(i-1)}}})\} \\
&\leq (1-\rho)^{\frac{k-1}{2}}(u_F - l_F) + 4C_1\{\sum_{i=k/2}^{k+1}\sqrt{\frac{\log m^{(i-1)}}{m^{(i-1)}}} \\
&\quad + \sum_{i=1}^{k/2-1}(1-\rho)^{k/2-i}\sqrt{\frac{\log m^{(i-1)}}{m^{(i-1)}}}\} \\
&\qquad\qquad (\tfrac{\log x}{x} \text{ decreases for } x > e) \\
&\leq (1-\rho)^{\frac{k-1}{2}}(u_F - l_F) + 4C_1\sqrt{2\log\frac{1}{1-\rho}}\{ \\
&\quad \sum_{i=k/2}^{k+1}(1-\rho)^i\sqrt{i} + \sum_{i=1}^{k/2-1}(1-\rho)^{k/2}\sqrt{i}\} \\
&\leq C_2(1-\beta)^{\frac{k-1}{2}}
\end{aligned}
$$

for some constant C_2 and $0 < \beta < \rho < 1$. □

Remark: The proof of Theorem 1 does not use any stochastic arguments until the very end, where it uses Lemma 4 to get rid of the indicator function for the 'drop-steps' so that the stochastic arguments based on concentration inequalities can be applied. Note that we cannot take expectation on the stochastic gradients and utilize their unbiasedness property because of the presence of the indicator functions. This proof technique is specifically designed for the 'drop-step' in ASFW and can be useful in analyzing other similar algorithms.

Corollary 2. *Let $\{\mathbf{x}^{(k)}\}_{k\geq 1}$ be the sequence generated by Algorithm 2 for solving Problem $(P1)$. Then*

$$
\frac{F(\mathbf{x}^{(k)}) - F^*}{(1-\omega)^{\frac{k-1}{2}}} \to 0
$$

almost surely as k tends to infinity for some $0 < \omega < \beta$. Therefore $F(\mathbf{x}^{(k)})$ linearly converges to F^* almost surely.

Proof. For every $\epsilon > 0$, let $E^{(k)}$ denotes the event that $(F(\mathbf{x}^{(k)}) - F^*)/(1-\omega)^{(k-1)/2} > \epsilon$. By Markov inequality

$$
\begin{aligned}
\sum_{k=2}^{\infty}\mathbb{P}(E^{(k)}) &= \sum_{k=1}^{\infty}\mathbb{P}((F(x^{(k)}) - F^*)/(1-\omega)^{(k-1)/2} > \epsilon) \\
&\leq \sum_{k=2}^{\infty}\frac{\mathbb{E}\{F(x^{(k)}) - F^*\}}{\epsilon(1-\omega)^{(k-1)/2}} \\
&\leq \frac{C_2}{\epsilon}\sum_{k=2}^{\infty}(\frac{1-\beta}{1-\omega})^{\frac{k-1}{2}} < \infty.
\end{aligned}
$$

Therefore $\sum_{k=2}^{\infty}\mathbb{P}(E^{(k)}) < \infty$ and the Borel-Cantelli lemma implies that $\mathbb{P}(\limsup_{k\to\infty} E^{(k)}) = 0$ which implies $(F(\mathbf{x}^{(k)}) - F^*)/(1-\omega)^{(k-1)/2}$ converges to 0 almost surely. This implies that every sequence generates by Algorithm 2 linearly converges to the optimal function value almost surely. □

Remark: Note that the result in Corollary 2 only relies on the property that an algorithm converges linearly in expectation. Therefore, we can apply exactly the same argument to show that every sequence generated by the algorithm in Johnson and Zhang [2013] converges linearly almost surely.

Corollary 3. *To obtain an ϵ-accurate solution, Algorithm 2 requires $O((1/\epsilon)^{4\eta})$ of stochastic gradient evaluations, where $0 < \eta = \log(1-\rho)/\log(1-\beta) < 1$.*

Theorem 2. *Let $\{\mathbf{x}^{(k)}\}_{k\geq 1}$ be the sequence generated by Algorithm 3 for solving Problem $(P1)$, N be the number of vertices used to represent $\mathbf{x}^{(k)}$ (if VRU is implemented by using Carathéodory's theorem, $N = p+1$, otherwise $N = |V|$) and F^* be the optimal value of the problem. Let $\kappa = \min\{\frac{1}{2}, \frac{\Omega_P^2\sigma_F}{8N^2L_FD^2}\}$ where $\sigma_F = \min\{\sigma_1, \ldots, \sigma_n\}$, $L_F = \max\{L_1, \ldots, L_n\}$. Set $m^{(i)} = \lceil 1/(1-\kappa)^{2i+2}\rceil$. Then for every $k \geq 1$*

$$
\mathbb{E}\{F(\mathbf{x}^{(k+1)}) - F^*\} \leq C_3(1-\phi)^{k/(3|V|!+1)} \qquad (4)
$$

where C_3 is a deterministic constant and $0 < \phi < \kappa \leq 1/2$.

Proof of Corrolary 3 and Theorem 2 are given in the supplementary material.

Corollary 4. *Let $\{\mathbf{x}^{(k)}\}_{k\geq 1}$ be the sequence generated by Algorithm 3 for solving Problem $(P1)$. Then*

$$\frac{F(\mathbf{x}^{(k)}) - F^*}{(1-\psi)^{\frac{k}{3|V|!+1}}} \to 0$$

almost surely as k tends to infinity for some $0 < \psi < \phi$. Therefore $F(\mathbf{x}^{(k)})$ linearly converges to F^* almost surely.

Corollary 5. *To obtain an ϵ-accurate solution, Algorithm 3 requires $O((1/\epsilon)^{(6|V|!+2)\xi})$ of stochastic gradient evaluations, where $0 < \zeta = \log(1-\rho)/\log(1-\phi) < 1$.*

Proof of Corollary 5 is the same as the proof of Corollary 3 and proof of Corollary 4 is almost the same as the proof of Corollary 2.

Remark: If we implement step 18 (Carathéodory's theorem), there will be additional computational cost of $O(p^2)$ operations per iteration. Details can be found in Beck and Shtern [2015].

Remark: The proof technique also works for general stochastic optimization problems with appropriate assumptions.

4 NUMERICAL EXPERIMENTS

4.1 Simulated Data

We apply the proposed algorithms to the synthetic problem below:

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \frac{1}{2}\|\mathbf{x}\|_2^2 \\ \text{such that} \quad & l \leq x_1 \leq x_2 \leq \cdots \leq x_p \leq u, \end{aligned}$$

where $\mathbf{A} \in \mathbb{R}^{n\times p}$, $\mathbf{b} \in \mathbb{R}^n$ and $\mathbf{x} \in \mathbb{R}^p$. We generated the entries of $\mathbf{A}$ and $\mathbf{b}$ from standard normal distribution and set $n = 10^6$, $p = 1000$, $l = -1$ and $u = 1$. This problem can be viewed as minimizing a sum of strongly convex functions subject to a polytope constraint. Such problems can be found in the shape restricted regression literature. We compared ASFW and PSFW with two variance-reduced stochastic methods, the variance-reduced stochastic Frank-Wolfe (SVRF) method [Luo and Hazan, 2016] and the proximal variance-reduced stochastic gradient (Prox-SVRG) method [Johnson and Zhang, 2013] [Lin and Zhang, 2014]. Both Prox-SVRG and SVRF are epoch based algorithms. They first fix a reference point and compute the exact gradient at the reference point at the beginning of each epoch. Within each epoch, both algorithms compute variance reduced gradients in every step using the technique of control variates based on the reference point. The major difference between them is that in every iteration, the Prox-SVRG takes a proximal gradient step and the SVRF takes a Frank-Wolfe step. For detailed implementations of SVRF, we followed Algorithm 1 in Luo and Hazan [2016] and chose the parameters according to Theorem 1 in Luo and Hazan [2016]. For the Prox-SVRG, we followed the Algorithm in Lin and Zhang [2014] and set the number of iterations in each epoch to be $m = 2n$ and set the step size to be $\gamma = 0.1/L$ found by Lin and Zhang [2014] to give the best results for Prox-SVRG, where n is the sample size and L is the Lipschitz constant for the gradient of the objective function. For the ASFW and PSFW implementations, we followed Algorithm 2 and Algorithm 3 and used adaptive step sizes since we know the Lipschitz constant of the gradient of the objective function. The number of samples that we used to compute stochastic gradients for ASFW and PSFW was set to be $1.04^k + 100$ at iteration k. The linear optimization sub-problems in Frank-Wolfe algorithms and the projection step in Prox-SVRG were solved by using the GUROBI solver. We summarize the parameters that were used in the algorithms at iteration k and epoch t in Table 2.

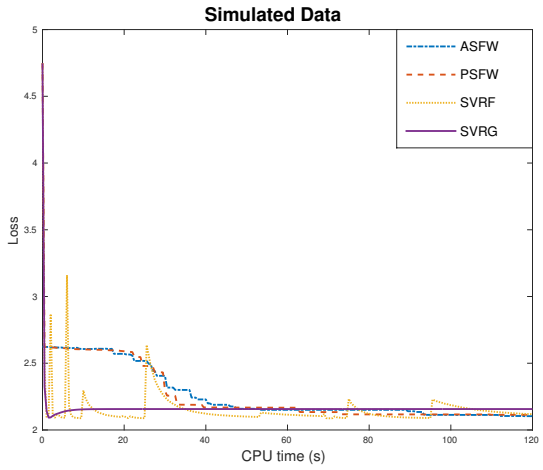

Figure 1: Loss vs cpu time for the synthetic problem.

To make fair comparisons, we used the same starting point for all four algorithms. The loss functions for all four algorithms are plotted against CPU time. From the plot, we can see that ASFW and PSFW performed as well as or slightly better than their stochastic competitors. At the very beginning, Prox-SVRG has a more rapid descent, while ASFW and PSFW obtains smaller function values later on. We also observed big jumps in SVRF periodically. This is because at the beginning of each epoch, SVRF proceeds with noisy gradients and very large step sizes. According to Theorem 1 in Luo and Hazan [2016], the step size of the first step in every epoch can be as large as 1.

Table 2: Parameters used in all four algorithms

	step-size	batch-size	#iterations
ASFW	$\min\{-\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle/(L^{(k)}\|\mathbf{d}^{(k)}\|^2), \gamma_{\max}\}$	$100 + 1.04^k$	N/A
PSFW	$\min\{-\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)}\rangle/(L^{(k)}\|\mathbf{d}^{(k)}\|^2), \gamma_{\max}\}$	$100 + 1.04^k$	N/A
SVRF	$2/(k+1)$	$96(k+1)$	$2^{t+3} - 2$
SVRG	$0.1/L$	1	$2n$

In ASFW and PSFW, $\mathbf{g}^{(k)}$ is the stochastic gradient, $L^{(k)}$ is the Lipschitz constant of the stochastic gradient at iteration k, $\mathbf{d}^{(k)}$ is the direction the algorithms take at iteration k and $\gamma_{\max}$ is the maximum of the possible step sizes (see Algorithm 2 and 3). In Prox-SVRG, L is the Lipschitz constant of the gradient of the objection function and n is the sample size.

4.2 Million Song Dataset

We implemented ASFW and PSFW for solving least squares problems with elastic-net regularization and tested them on the Million Song Dataset (YearPredictionMSD) [Lichman, 2013][Bertin-Mahieux et al., 2011], which is a dataset of songs with the goal of predicting the release year of a song from its audio features. There are $n = 463,715$ training samples and $p = 90$ features in this dataset. The dataset is the one with largest number of training samples available in the UCI machine learning data repository. Therefore it is interesting to examine the actual performance of stochastic algorithms on such a massive dataset. The least squares with elastic-net regularization model that we used was,

$$\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{n}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1 + \mu\|\mathbf{x}\|_2^2$$

where $\mathbf{A} \in \mathbb{R}^{n\times p}$ and $\mathbf{b} \in \mathbb{R}^n$. $\mu \geq 0$ and $\lambda \geq 0$ are regularization parameters. In the numerical experiments, we considered the constrained version of the problem, that is,

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \mu\|\mathbf{x}\|_2^2 \\ \text{subject to} \quad & \|\mathbf{x}\|_1 \leq \alpha \end{aligned}$$

where $\alpha > 0$ is inversely related to λ. We also compared the ASFW and PSFW with SVRF and Prox-SVRG. We followed the same settings in this real data experiment as that in the simulated data experiment except that we used explicit solutions for solving linear optimizations over an l_1-balls in FW algorithms and we used the algorithm in Duchi et al. [2008] for the solving projections onto l_1-balls in the Prox-SVRG algorithm instead of using GUROBI for solving linear optimizations and projections. To make fair comparisons, we used the same starting point for all four algorithms. The logarithm of the loss functions obtained by ASFW, PSFW and Prox-SVRG and the running minimum obtained by SVRF are plotted against CPU time. The figures show that ASFW and PSFW performed better than Prox-SVRG and SVRF for all different regularization parameters. Please find more plots in the supplementary material. We also observed huge swings in SVRF in these experiments. Therefore we plotted the running minimums instead of the most recent function values for SVRF. We also note that SVRF appears unable to reduce the function to its minimum value for $\alpha \geq 1$.

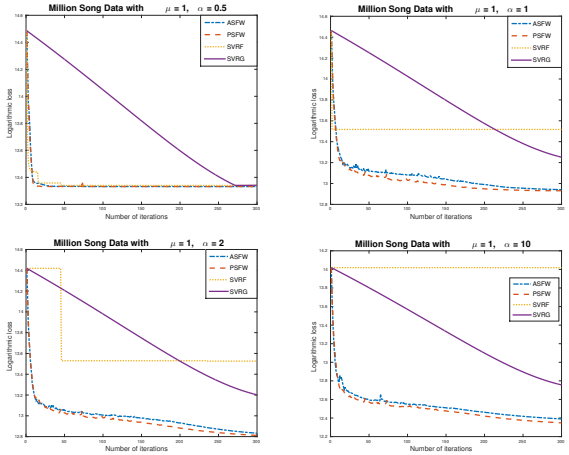

Figure 2: Logarithmic loss vs number of iterations on the elastic net problem with different parameter choices for Million Song dataset.

5 CONCLUSION AND FUTURE WORK

In this paper, we proved linear convergence almost surely and in expectation of the Away-step Stochastic Frank-Wolfe algorithm and the Pairwise Stochastic Frank-Wolfe algorithm by using a novel proof technique. We tested these algorithms by training a least squares model with elastic-net regularization on the million song dataset and on a synthetic problem. The proposed algorithms performed as well as or better than their stochastic competitors for various choice of the regularization parameters. Future work includes extending the proposed algorithms to problems with block-coordinate structures and non-strongly convex objective functions and using variance reduced stochastic gradients to reduce the number of stochastic gradient oracle calls.

References

J. Abadie, editor. *Integer and Nonlinear Programming*. North-Holland, Amsterdam, 1970.

A. Beck and S. Shtern. Linearly convergent away-step conditional gradient for non-strongly convex functions. *arXiv: 1504.05002*, 2015.

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3: 95110, 1956.

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

D. Garber and E. Hazan. Faster rates for the Frank-Wolfe method over strongly-convex sets. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, page 541549, 2015.

J. Guelat and P. Marcotte. Some comments on Wolfe's 'away step'. *Mathematical Programming*, 35(1):110119, 1986.

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

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 315–323, 2013.

S. Lacoste-Julien and M. Jaggi. An affine invariant linear convergence analysis for Frank-Wolfe algorithms. *arXiv:1312.7864v2*, 2014.

S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate Frank-Wolfe optimization for structural svms. *In Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 28:5361, 2013.

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

J. Lafond, H. Wai, and E. Moulines. Convergence analysis of a stochastic projection-free algorithm. *arXiv:1510.01171*, 2015.

G. Lan. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv:1309.5550*, 2013.

E. Levitin and B. T. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6(5):787823, 1966.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

X. Lin and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

H. Luo and E. Hazan. Variance-reduced and projection-free stochastic optimization. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, volume 48, 2016.

C. Mu, Y. Zhang, J. Wright, and D. Goldfarb. Scalable robust matrix recovery: Frank-Wolfe meets proximal methods. *arXiv:1403.7588*, 2015.

Yu. Nesterov. Complexity bounds for primal-dual methods minimizing the model of objective function. Technical report, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2015.

A. Osokin, J. B. Alayrac, I. Lukasewitz, P. K. Dokania, and S. Lacoste-Julien. Minding the gaps for block Frank-Wolfe optimization of structured svms. *arXiv:1605.09346*, 2016.

H. Ouyang and A. Gray. Fast stochastic Frank-Wolfe algorithms for nonlinear SVMs. In *SDM*, page 245256, 2010.

Aad W. van der Vaart and Jon A. Wellner, editors. *Weak Convergence and Empirical Processes With Applications to Statistics*. Springer, 1996.