Generalized Stochastic Frank-Wolfe Algorithm with Stochastic "Substitute" Gradient for Structured Convex Optimization

Haihao Lu[*]   Robert M. Freund[†]

July 2, 2019

Abstract

The stochastic Frank-Wolfe method has recently attracted much general interest in the context of optimization for statistical and machine learning due to its ability to work with a more general feasible region. However, there has been a complexity gap in the guaranteed convergence rate for stochastic Frank-Wolfe compared to its deterministic counterpart. In this work, we present a new generalized stochastic Frank-Wolfe method which closes this gap for the class of structured optimization problems encountered in statistical and machine learning characterized by empirical loss minimization with a certain type of "linear prediction" property (formally defined in the paper), which is typically present loss minimization problems in practice. Our method also introduces the notion of a "substitute gradient" that is a not-necessarily-unbiased sample of the gradient. We show that our new method is equivalent to a particular randomized coordinate mirror descent algorithm applied to the dual problem, which in turn provides a new interpretation of randomized dual coordinate descent in the primal space. Also, in the special case of a strongly convex regularizer our generalized stochastic Frank-Wolfe method (as well as the randomized dual coordinate descent method) exhibits linear convergence. Furthermore, we present computational experiments that indicate that our method outperforms other stochastic Frank-Wolfe methods consistent with the theory developed herein.

1 Introduction

Our problem of interest is the following optimization problem:

$$(\text{P}): \qquad \min_{\beta} P(\beta) := \tfrac{1}{n} \sum_{j=1}^{n} l_j(x_j^T \beta) + R(\beta) \ , \qquad (1)$$

where $\beta \in \mathbb{R}^p$, $l_j(\cdot) : \mathbb{R} \to \mathbb{R}$, $j = 1, \ldots, n$, is a univariate function (the jth loss function), $s_j = x_j^T \beta$ is the "predicted value" of the model β for the data sample x_j, and $R(\cdot)$ is some other function that can be used to model a variety of properties such as a regularizer, an indicator function of a feasible region Q, a penalty term, coupling constraints, etc. Notice that the predicted value $s_j = x_j^T \beta$ is a linear function of β. Hence in the intentional context of empirical loss minimization

---

we refer to (1) as (empirical) loss minimization with "linear prediction." This structure arises quite generally throughout statistical and machine learning, which we illustrate below via several salient examples. Throughout this paper we assume the following regarding the functions in our problem setup (1):

Assumption 1.1. *The following hold:*

1. *for $j = 1, \ldots, n$, the univariate function $l_j(\cdot)$ is γ-smooth, namely $|\dot{l}_j(a) - \dot{l}_j(b)| \leq \gamma|a - b|$ for all a, b, and is strictly convex,*

2. *$\mathrm{dom}R(\cdot)$ is bounded, and the subproblem*

$$\min_{\beta} c^T\beta + R(\beta) \tag{2}$$

   attains its optimum and can be easily solved for any c, and

3. *$0 \in \mathrm{dom}R(\cdot)$.*

We note regarding *(1.)* above that strict convexity (instead of simple convexity) is only needed to guarantee that the conjugate function $l_j^*(\cdot)$ is differentiable. Regarding *(2.)*, this is a generalization of a linear optimization oracle as follows: in the case when $R(\cdot)$ is the indicator function $\mathbf{I}_Q(\cdot)$ of a set $Q \subset \mathbb{R}^p$ (namely, $\mathbf{I}_Q(\beta) := 0$ if $\beta \in Q$, and $\mathbf{I}_Q(\beta) := +\infty$ otherwise), then Q is the feasible region of (P), and *(2.)* states that the feasible region Q is bounded and that it is easy to solve linear optimization problems on Q. Also, *(3.)* above is for notational convenience, as we can always translate a given feasible point so that $0 \in \mathrm{dom}R(\cdot)$.

1.1 Examples in Statistical and Machine Learning

Here we present several applications of our problem setup (1) in statistical and machine learning. (For other applications particularly amenable to solution by the Frank-Wolfe method, we refer the reader to [19].)

Example 1.1. LASSO[41], ridge regression[18], sparse logisitic regression[34]. *Consider the least-squares regression problem where a set of training samples $\{(x_j, y_j)\}_{j=1}^n$ is given. The LASSO optimization problem (in constraint format) is:*

$$\min_\beta \quad \frac{1}{2n} \sum_{j=1}^n (y_j - x_j^T\beta)^2$$

$$s.t. \quad \|\beta\|_1 \leq \delta \ ,$$

which is an instance of (P) by using the least squares loss function $l_j(\cdot) = \frac{1}{2}(y_j - \cdot)^2$ and using the indicator function of an ℓ_1 ball as the "regularizer" function $R(\cdot)$, namely $R(\beta) := \mathbf{I}_{\{\|\beta\|_1 \leq \delta\}}(\beta)$.

The ridge regression optimization problem adds the regularizer $\frac{\lambda}{2}\|\beta\|_2^2$ to the least squares objective function for the parameter $\lambda > 0$, and omits the ℓ_1 ball constraint. Notice that because $\beta = 0$ is a feasible solution it follows that the optimal objective value is bounded above by $\|y\|_2^2/(2n)$, and therefore we can model $R(\beta) = \frac{\lambda}{2}\|\beta\|_2^2 + \mathbf{I}_{\{\|\beta\|_2^2 \leq \|y\|_2^2/(n\lambda)\}}(\beta)$, which ensures that $\mathrm{dom}R(\cdot)$ is bounded.

2

The ℓ_1-regularized logistic regression optimization problem seeks a solution of:

$$\min_{\beta} P(\beta) = \frac{1}{n} \sum_{j=1}^{n} \ln(1 + \exp(-y_j x_j^T \beta)) + \lambda \|\beta\|_1 ,$$

for a given set of training samples $\{(x_j, y_j)\}_{j=1}^n$ where $y_j \in \{-1, 1\}$, and is an instance of (P) using the logistic loss functions $l_j(\cdot) = \ln(1 + \exp(-y_j \cdot))$ with $R(\beta) = \lambda \|\beta\|_1 + \mathbf{I}_{\{\|\beta\|_1 \leq \ln(2)/\lambda\}}(\beta)$ where the indicator function term is structurally redundant but is added as in the previous example to ensure that $\mathrm{dom} R(\cdot)$ is bounded.

Example 1.2. Matrix completion[10][7]. *In the matrix completion problem, we seek to compute a low-rank matrix that well-approximates a given matrix $M \in \mathbb{R}^{n \times p}$ on the set Ω of observed entries (i, j). The convex relaxation of this problem is the following nuclear-norm optimization problem:*

$$\min_{\beta \in \mathbb{R}^{n \times p}} \quad \frac{1}{2|\Omega|} \sum_{(i,j) \in \Omega} (M_{i,j} - \beta_{i,j})^2$$

$$s.t. \qquad \|\beta\|_* \leq \delta ,$$

where $\|\cdot\|_*$ is the nuclear norm. In order to translate the matrix completion problem to the setting of (P), we consider any index pair $(i, j) \in \Omega$ as a sample, and we have $l_{(i,j)}(\cdot) = \frac{1}{2}(\cdot - M_{i,j})^2$, and $R(\beta) = \mathbf{I}_{\{\|\beta\|_* \leq \delta\}}(\beta)$.

Example 1.3. Structured sparse matrix estimation with CUR factorization[27][29]. *We seek to compute an approximate factorization $M \approx CUR$ of a given data matrix $M \in \mathbb{R}^{n \times d}$ such that C contains a subset of c columns from M and R contains a subset of r rows from M. Mairal et al. [29] proposed the following convex relaxation of this problem:*

$$\min_{\beta} \quad \frac{1}{2nd} \|M - M\beta M\|_F^2$$

$$s.t. \quad \sum_i \|\beta_{i,\cdot}\|_\infty \leq \delta$$

$$\sum_j \|\beta_{\cdot,j}\|_\infty \leq \delta ,$$

which is an instance of (P) by modeling the $(i, j)^{\mathrm{th}}$ loss term in (P) as $\frac{1}{2}(M_{i,j} - M_i^T \beta M_j)^2$ (which is a least squares loss of a particular linear function of the matrix variable β), and $R(\beta) = \mathbf{I}_{\{\sum_i \|\beta_{i,\cdot}\|_\infty \leq \delta, \ \sum_j \|\beta_{\cdot,j}\|_\infty \leq \delta\}}(\beta)$.

1.2 Stochastic Generalized Frank-Wolfe with Stochastic Substitute Gradient

Let $X \in \mathbb{R}^{n \times p}$ denote the data matrix whose rows are comprised of the vectors $x_1, \ldots, x_n$, i.e., the jth row of X is the vector x_j, $j = 1, \ldots, n$. Let us define $L(s) : \mathbb{R}^n \to \mathbb{R}$ by $L(s) := \sum_{j=1}^{n} l_j(s_j)$ which is the total losses associated with $s \in \mathbb{R}^n$. We refer to $s = X\beta$ as the "predicted values" in the context of empirical loss minimization.

Algorithm 1 presents the main algorithmic contribution of this paper, which is a first-order method for tackling the problem (P). We call the method "Stochastic Generalized Frank-Wolfe method with Stochastic Substitute Gradient" for reasons which we will discuss as we walk through the structure of the method below.

3

Algorithm 1 Stochastic Generalized Frank-Wolfe with Stochastic Substitute Gradient

---

Initialize. Initialize with $\bar{\beta}^{-1} = 0$, $s^0 = 0$, and substitute gradient $d^0 = \frac{1}{n}X^T\nabla L(s^0)$, with step-size sequences $\{\alpha_i\} \in (0,1]$ and $\{\eta_i\} \in (0,1]$.

For iterations $i = 0, 1, \ldots$

      Solve l.o.o. subproblem: Compute $\tilde{\beta}^i \in \arg\min_\beta \left\{ \left(d^i\right)^T \beta + R(\beta) \right\}$

      Choose random index: Choose $j_i \in \mathcal{U}[1,\ldots,n]$

      Update s value: $s_{j_i}^{i+1} \leftarrow (1-\eta_i)s_{j_i}^i + \eta_i(x_{j_i}^T\tilde{\beta}^i)$, and $s_j^{i+1} \leftarrow s_j^i$ for $j \neq j_i$

      Update substitute gradient: $d^{i+1} = \frac{1}{n}X^T\nabla L(s^{i+1}) = d^i + \frac{1}{n}\left(\dot{l}_{j_i}(s_{j_i}^{i+1}) - \dot{l}_{j_i}(s_{j_i}^i)\right)x_{j_i}$

      Update primal variable: $\bar{\beta}^i \leftarrow (1-\alpha_i)\bar{\beta}^{i-1} + \alpha_i\tilde{\beta}^i$.

      (Optional Accounting:) $w^{i+1} \leftarrow \nabla L(s^{i+1})$

We can write the first part of the objective function of (P) as $f(\beta) := \frac{1}{n}L(X\beta) = \frac{1}{n}L(s)$ with $s = X\beta$. We have $\nabla L(s) = (\dot{l}_1(s_1), \ldots, \dot{l}_n(s_n))$ and the gradient of $f(\cdot)$ can be written as

$$\nabla f(\beta) = \tfrac{1}{n}X^T\nabla L(X\beta) = \tfrac{1}{n}\sum_{j=1}^n \dot{l}_j(x_j^T\beta)x_j \ , \tag{3}$$

which we can re-write as $\nabla f(\beta) = \frac{1}{n}X^T w$ where $w = \nabla L(s)$ and $s = X\beta$, and which can be alternatively stated as:

$$\nabla f(\beta) = \tfrac{1}{n}\sum_{j=1}^n w_j x_j \text{ where } w_j = \dot{l}_j(s_j) \text{ and } s_j = x_j^T\beta \ , \ j = 1, \ldots, n \ . \tag{4}$$

Here we emphasize that w is the vector of weights on the data values X in the composition of the gradient, and s is the vector of predicted values $X\beta$.

Especially in the context of "big data" applications of statistical and machine learning where n is huge, it can be extremely expensive to compute $\nabla f(\cdot)$. We therefore maintain a "substitute gradient" in Algorithm 1 that is constructed stochastically. This is accomplished as follows: let $\bar{\beta}^{i-1}$ be the value of β at the start of iteration i of the method, and we have a substitute gradient d^i that is the current proxy/substitute for the true gradient $\nabla f(\bar{\beta}^{i-1})$, where d^i is computed by:

$$d^i = \tfrac{1}{n}\sum_{j=1}^n w_j^i x_j \text{ where } w_j^i = \dot{l}_j(s_j^i) \ , \ j = 1, \ldots, n \ , \tag{5}$$

for a given s^i that is the value of s at iteration i. But in contrast to (4) it will <u>not</u> necessarily hold that $s_j^i = x_j^T\bar{\beta}^i$ for $j = 1, \ldots, n$ (equivalently $s^i = X\bar{\beta}^i$). (In fact, d^i will not necessarily be an unbiased estimate of $\nabla f(\bar{\beta}^i)$ as this will not be needed.) In the identical spirit as randomized coordinate descent, s^{i+1} will be determined by choosing a random index $j_i \in \mathcal{U}[1,\ldots,n]$ and updating only the coordinate j_i of s^i, so that $s^{i+1} = s^i + \Delta^i e_{j_i}$ for some specific iteration-dependent scalar Δ^i (where e_ℓ denotes the ℓth unit coordinate vector in $\mathbb{R}^n$). This is accomplished in the "choose random index" step and the "update s value" step in Algorithm 1.

4

We now walk through the structure of Algorithm 1 in complete detail. The method is initialized with the initial decision variable β set to $\bar{\beta}^{-1} = 0$ and the vector of predicted values $s^0 = X\bar{\beta}^{-1} = 0$ and initial substitute gradient $d^0 = \frac{1}{n}X^T\nabla L(s^0)$, which corresponds to the true predicted value and true gradient at $\bar{\beta}^{-1} = 0$. In iteration i, we use the substitute gradient d^i to compute $\tilde{\beta}^i$, which is a solution to the (generalized) linear optimization oracle ("l.o.o."), where recall that this step specifies to solving a linear optimization problem over a set Q in the specific case when the $R(\cdot)$ is the indicator function of Q, namely $R(\cdot) = \mathbf{I}_Q(\cdot)$. Regarding updating the current predicted values s^i, we randomly choose a sample (a coordinate) j_i and only update s_{j_i} as a certain convex combination of the current predicted value $s^i_{j_i}$ and the predicted value for the j_i^{th} sample at $\tilde{\beta}^i$, namely $x_{j_i}^T\tilde{\beta}^i$, so that $s^{i+1}_{j_i} \leftarrow (1-\eta_i)s^i_{j_i} + \eta_i(x_{j_i}^T\tilde{\beta}^i)$. Then we update the substitute gradient to make sure that $d^{i+1} = \frac{1}{n}X^T\nabla L(s^{i+1})$. The last step at iteration i is to take a Frank-Wolfe step to update $\bar{\beta}^i \leftarrow (1-\alpha_i)\bar{\beta}^{i-1} + \alpha_i\tilde{\beta}^i$ by taking a convex combination of the previous primal variable value $\bar{\beta}^{i-1}$ and the solution $\tilde{\beta}^i$ of the just-solved linear optimization oracle. Finally – and "optionally" since it does not affect future computations – we can perform an optional accounting step to update the dual variable $w^{i+1} \leftarrow \nabla L(s^{i+1})$ in order to compute a duality gap certificate if desired. (The nature of this duality will be understood once we look at the dual problem of (P) in Section 2.)

Note that the computations in Algorithm 1 are minimally affected by the dimension n. Except for the initial computation of the gradient d^0 which is $O(np)$ operations, s^i and w^i are only updated by one coefficient at each iteration, and d^{i+1} is updated by adding a scalar multiple of x_{j_i} to d^i, which is $O(p)$ operations. The updates of $\bar{\beta}^i$ are $O(p)$ operations after solving for the optimal value $\tilde{\beta}^i$ in the linear optimization oracle, which is assumed to be easy to compute.

It is useful to place Algorithm 1 in the context of the (deterministic) Frank-Wolfe method. The Frank-Wolfe method is designed primarily to tackle the constrained convex optimization problem: $\min_{\beta \in Q} f(\beta)$ where $f(\cdot)$ is a smooth convex function and Q is a convex body, and it is assumed that linear optimization over Q is easy to compute. The optimization problem can of course be re-written as $\min_\beta f(\beta) + R(\beta)$ with $R(\cdot) = \mathbf{I}_Q(\cdot)$. The Frank-Wolfe update is:

$$\tilde{\beta}^i \in \arg\min_{\beta \in Q}\left\{\nabla f(\beta^i)^T\beta\right\} \quad \text{and} \quad \beta^{i+1} = (1-\alpha_i)\beta^i + \alpha_i\tilde{\beta}^i \ . \tag{6}$$

It can be shown that with an appropriate choice of step-size sequence $\{\alpha_i\}$ that the Frank-Wolfe method computes an ε-optimal solution in $O\left(\frac{LD^2}{\varepsilon}\right)$ iterations, where L is the Lipschitz constant of $f(\cdot)$ on Q and D is the diameter of Q, see [11], [19], and [12]. Since our focus (for stochastic versions of Frank-Wolfe) will be on the dependence on ε, we will typically ignore these other instance-dependent constants and write the above as $O(\frac{1}{\varepsilon})$ iterations.

Due to its low iteration cost and convenient structural properties, the Frank-Wolfe method is especially applicable in several areas of statistical and machine learning and has thus received much renewed interest in recent years, see [19], [15], [13], [12], and the references therein. The Frank-Wolfe method can be generalized to deal with the more general problem $\min_\beta f(\beta) + R(\beta)$ where $R(\cdot)$ is any convex function with bounded domain and for which the "linear optimization problem" $\min_\beta c^T\beta + R(\beta)$ is easy to compute. The generalized Frank-Wolfe update then is:

$$\tilde{\beta}^i \in \arg\min_\beta\left\{\nabla f(\beta^i)^T\beta + R(\beta)\right\} \quad \text{and} \quad \beta^{i+1} = (1-\alpha_i)\beta^i + \alpha_i\tilde{\beta}^i \ , \tag{7}$$

and notice that we recover the regular Frank-Wolfe update in the special case when $R(\cdot)$ is the

indicator function $\mathbf{I}_Q(\cdot)$ of a feasible region Q, see [4] and [45] for a more detailed discussion on generalized Frank-Wolfe methods.

1.3   Related literature

Stochastic Frank-Wolfe methods. There have been several lines of research that develop and investigate stochastic Frank-Wolfe methods, almost all of which are motivated by expected loss minimization in statistical and machine learning. These methods by and large focus on the optimization problem:

$$\min_{\beta \in Q} f(\beta) = \tfrac{1}{n} \sum_{i=1}^{n} f_j(\beta) \ , \tag{8}$$

where Q is a closed and bounded convex set. Here $f(\cdot)$ is the empirical risk, though some of the results we discuss below pertain to the more general "infinite" setting of expected loss minimization where

$$f(\beta) = \int_{\xi} f(\beta; \xi) d\mu(\xi) \tag{9}$$

for some appropriate probability measure $\mu(\cdot)$.

Hazan and Luo [17] discuss several different stochastic Frank-Wolfe algorithms with increasing batch sizes over the course of iterations, including a straightforward stochastic Frank-Wolfe method (SFW), a stochastic variance reduced Frank-Wolfe method (SVRFW), and a stochastic variance-reduced conditional gradient sliding method (STORC). In order to compute an ε-optimal solution, SFW requires $O(\frac{1}{\varepsilon^3})$ stochastic gradient calls (i.e., one computation of the gradient of one of the $f_j(\cdot)$ above) and $O(\frac{1}{\varepsilon})$ linear optimization oracle calls. (Recall that we ignore other instance-specific constants such as Lipschitz constants, curvature, and diameters constants, as our focus here is on the dependence on ε.) SVRFW needs $O(\frac{1}{\varepsilon^2})$ stochastic gradient calls and $O(\frac{1}{\varepsilon})$ linear optimization oracle calls, as well as $O(\ln(\frac{1}{\varepsilon}))$ full gradient calls. STORC (for STOchastic variance-Reduced Conditional gradient sliding) is a variance-reduced version of the conditional gradient sliding method of Lan and Zhou [22]. The number of stochastic gradient calls that STORC needs is a function of some instance-specific properties: it is $O(\frac{1}{\varepsilon})$ if there is an optimal solution in the interior of Q, it is $O(\frac{1}{\varepsilon^{1.5}})$ more broadly, and is $O(\ln(\frac{1}{\varepsilon}))$ under strong convexity of $f(\cdot)$. STORC also needs $O(\frac{1}{\varepsilon})$ linear optimization oracle calls as well as $O(\ln(\frac{1}{\varepsilon}))$ full gradient calls.

Lan and Zhou [22] present the stochastic conditional gradient sliding (SCGS) algorithm, which combines Nesterov's acceleration techniques and the Frank-Wolfe method. In the absence of strong convexity, their stochastic Frank-Wolfe methodology requires $O(\frac{1}{\varepsilon^2})$ stochastic gradient calls and $O(\frac{1}{\varepsilon})$ linear optimization oracle calls. And in the presence of strong convexity, their method requires $O(\frac{1}{\varepsilon})$ stochastic gradient calls and $O(\frac{1}{\varepsilon})$ linear optimization oracle calls.

Mokhtari, Hassani and Karbasi [30] propose a stochastic Frank-Wolfe method called SCG (for stochastic conditional gradient) by introducing a momentum gradient estimator which does not require to increase the batch size. SCG requires $O(\frac{1}{\varepsilon^3})$ stochastic gradient calls and $O(\frac{1}{\varepsilon^3})$ linear optimization oracle calls.

In the online setting, Hazan and Kale [16] proposed an online Frank-Wolfe method which requires $O(\frac{1}{\varepsilon^4})$ stochastic gradient calls and $O(\frac{1}{\varepsilon^4})$ linear optimization oracle calls.

Of the above mentioned methods, we point out as well that the algorithms SFW [17] as well as SCGS [22] and SCG [30] can also be implemented in the "infinite" setting of (9).

Unlike the deterministic Frank-Wolfe method, the above stochastic Frank-Wolfe methods do not achieve $O(\frac{1}{\varepsilon})$ gradient calls and $O(\frac{1}{\varepsilon})$ linear optimization oracle calls to achieve an ε-optimal solution, without some additional restrictive assumptions. (This is the "complexity gap" mentioned in the abstract.) From the above discussion, we see that such complexity is achieved only for STORC [17] (when either (i) an optimal solution lies in the interior of Q, or (ii) when $f(\cdot)$ is strongly convex) and for SCGS [22] when $f(\cdot)$ is strongly convex.

The primary motivation (and contribution) of this paper is to show that the typical setting of empirical loss minimization – in which linear prediction is present – is sufficient to guarantee that a suitable stochastic Frank-Wolfe method achieves both $O(\frac{1}{\varepsilon})$ gradient calls and $O(\frac{1}{\varepsilon})$ linear optimization oracle calls to achieve an ε-optimal solution. Stated in the context of the literature on this topic, one does not need strong convexity nor interior optimal solutions to achieve the same complexity as deterministic Frank-Wolfe, so long as the setting is empirical loss minimization with linear prediction – which is quite prevalent in such models.

Randomized Dual Coordinate Descent Methods. One of the interpretations of our stochastic Frank-Wolfe method is that it is a dual coordinate descent method in the dual space. It thus is relevant to review the appropriate literature on dual coordinate descent in this context. Dual coordinate descent methods have been widely used in statistical and machine learning applications. Stochastic dual coordinate ascent (SDCA) for solving (1) was first proposed in [39]. There are many follow-up works on SDCA, for example, accelerated proximal randomized dual coordinate, see [40], [23], using a non-uniform distribution to choose the coordinate [33], and a primal-dual coordinate method [47], among others. All of these dual methods (or primal-dual methods) require the regularizer $R(\cdot)$ to be a strongly convex function (or require adding a dummy strongly convex regularizer to the objective function). This contrasts with the standard Frank-Wolfe set-up where $R(\cdot)$ is an indicator function and so is not strongly convex. Furthermore, in the Frank-Wolfe setup, the objective function in the dual problem (10) is not necessarily differentiable, which falls outside of the standard set-up for randomized coordinate descent [35], [32]. (However, it turns out that as a byproduct of our analysis we obtain convergence guarantees for randomized coordinate descent applied to the (non-differentiable) dual problem (10).) We further discuss the connections and differences between the above methods and our method in Appendix A.4.

Variance Reduction Techniques for Stochastic Optimization. There have been many recent developments of stochastic methods designed to directly tackle the optimization problem (1). In order to obtain improved convergence guarantees over the standard Stochastic Gradient Descent (SGD) method, variance reduction techniques have been proposed and extensively studied in recent years. SAG [36] is the first variance reduction method in the literature that we are aware of. In contrast to the sublinear convergence rate of SGD, SAG and several concurrent and/or subsequent works – such as SVRG [20], MISO [28], and SAGA [9] – obtain linear convergence when the objective function is both smooth and strongly convex. Variance reduction techniques can also be applied to non-strongly convex optimization [36], [28], [9], [2], which leads to improved convergence guarantees as well. More recently, Allen-Zhu [1] has proposed an accelerated stochastic method for directly solving (1). We mention as well that the stochastic dual coordinate method [39] also corresponds to a variant of a variance reduction technique in the primal space [37]. We refer the reader to [1]

for a more detailed discussion on variance reduction techniques overall.

1.4   Contributions

The overall contribution of this paper is Algorithm 1, which is a generalized stochastic Frank-Wolfe method (hence the moniker GSFW) designed to solve the empirical risk minimization problem with linear prediction (1). The specific contributions of Algorithm 1 and its analysis are as follows:

1. GSFW requires $O(\frac{1}{\varepsilon})$ stochastic gradient oracle calls and $O(\frac{1}{\varepsilon})$ linear optimization oracle calls to compute an absolute ε-optimal solution of the empirical risk minimization problem with linear prediction (1), see Theorem 3.1. This in particular demonstrates that the typical setting of empirical loss minimization – in which linear prediction is present – is sufficient to guarantee that a suitable stochastic Frank-Wolfe method achieves the same complexity as its deterministic counterpart. Stated in the context of the prevalent literature on stochastic Frank-Wolfe, one does not need strong convexity nor interior optimal solutions to achieve the same complexity as deterministic Frank-Wolfe, so long as the setting is empirical loss minimization with linear prediction – which is quite prevalent in such models.

2. In the special case when $R(\cdot)$ is strongly convex, GSFW requires $O(\ln(\frac{1}{\varepsilon}))$ stochastic gradient oracle calls and $O(\ln(\frac{1}{\varepsilon}))$ linear optimization oracle calls to compute an absolute ε-optimal solution of (1), see Theorem 3.2.

3. We show that GSFW is equivalent to a randomized coordinate mirror descent algorithm applied to the dual problem (Algorithm 2), see Lemma 2.1. Algorithm 2 can be viewed as a variant of the SDCA algorithm, and in this lens our method can handle a non-strongly convex function $R(\cdot)$, in contrast to the current SDCA literature.

4. Our work also implies a convergence bound for randomized coordinate mirror descent in the case when the objective function is the sum of a non-smooth function and a strongly convex separable function. This is discussed in Section A.5.

5. The recognition of the empirical risk minimization problem with linear prediction (1) as a problem of special interest due to the linear prediction structure.

1.5   Notation

We use e_j to denote the jth unit coordinate vector in $\mathbb{R}^p$. The ℓ_p norm is denoted $\|\cdot\|_p$. We use $\dot{l}_j(\cdot)$ to denote the first derivative of $l_j(\cdot)$. The Bregman distance function associated with a convex function $h(\cdot)$ is defined as $D_h(y,x) := h(y) - h(x) - \nabla h(x)^T(y-x)$. We use $\mathbb{E}$ to denote expectation and $\mathbb{E}_{j_i}$ to denote expectation conditional on the randomly chosen index j_i. For indicator functions, we use $\mathbf{I}_Q(\cdot)$ to denote the indicator function for the set Q, namely $\mathbf{I}_Q(\beta) := 0$ if $\beta \in Q$, and $\mathbf{I}_Q(\beta) := +\infty$ otherwise; and we use $\mathbf{I}_{\{constraint\}}(\beta)$ to denote the indicator function of a particular constraint (or condition), namely $\mathbf{I}_{\{constraint\}}(\beta) := 0$ if the constraint is true at β, and $\mathbf{I}_{\{constraint\}}(\beta) := +\infty$ otherwise. In a slight abuse of terminology we refer to the "subgradient" of a concave function when it is perhaps more technically accurate to refer to this as a sup-gradient. A differentiable function $f(\cdot)$ is μ-strongly convex with respect to a norm $\|\cdot\|$ if it holds that $f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\mu}{2}\|y-x\|^2$ for all $x, y \in \text{dom} f(\cdot)$. A

differentiable function $f(\cdot)$ is μ-strongly convex with respect to a reference function $h(\cdot)$ if it holds that $f(y) \geq f(x) + \nabla f(x)^T (y - x) + \mu D_h(y, x)$ for all $x, y \in \mathrm{dom} f(\cdot)$.

2 Dual problem, and equivalence of Algorithm 1 in the dual with Randomized Coordinate Mirror Descent

Recall the definition of the conjugate of a function $f(\cdot)$:

$$f^*(y) := \sup_{x \in \mathrm{dom} f(\cdot)} \{y^T x - f(x)\} \ .$$

We will also be interested in the following dual problem of (1) that is constructed using the conjugate functions of the component functions of (1):

$$(\mathrm{D}) : \qquad \max_w D(w) := -R^* \left(-\tfrac{1}{n} X^T w\right) - \tfrac{1}{n} \sum_{j=1}^n l_j^*(w_j) \ . \tag{10}$$

Notice that we can write:

$$R^* \left(-\tfrac{1}{n} X^T w\right) = -\min_\beta \left\{\tfrac{1}{n} w^T X \beta + R(\beta)\right\} \ . \tag{11}$$

Also, defining the convex/concave saddle-function $\phi(\cdot, \cdot)$:

$$\phi(\beta, w) := \tfrac{1}{n} w^T X \beta - \tfrac{1}{n} \sum_{i=1}^n l_i^*(w_i) + R(\beta) \ , \tag{12}$$

we can write (P) and (D) in saddlepoint minimax format as:

$$(\mathrm{P}) : \quad \min_\beta \max_w \phi(\beta, w) \qquad \text{and} \qquad (\mathrm{D}) : \quad \max_w \min_\beta \phi(\beta, w) \ . \tag{13}$$

Another standard first-order method for convex optimization is the mirror descent algorithm (also called primal gradient method with Bregman distance) [42], [25], [24], [6], which we now briefly review in the context of solving the dual problem (D) in (10), which is a concave maximization problem. The Bregman distance of a differentiable "prox" function $h(\cdot)$ is defined to be:

$$D_h(w_1, w_2) := h(w_1) - h(w_2) - \langle \nabla h(w_2), w_1 - w_2 \rangle \ .$$

The (deterministic) mirror descent algorithm for solving (D) has the following update:

$$w^{i+1} \leftarrow \arg\min_w \{-\eta_i g(w^i)^T (w - w^i) + D_h(w, w^i)\} \ ,$$

where $g(\cdot)$ is a subgradient of the objective function $D(\cdot)$ at w (which we call a subgradient even though $D(\cdot)$ is concave), and $\{\eta_i\}$ is the step-size sequence. It is shown in Bach [4] that the generalized Frank-Wolfe method for the primal (1) is equivalent to mirror descent algorithm for the dual (10).

Algorithm 2 presents a Randomized Coordinate Mirror Descent method applied to solve the dual problem D. The algorithm uses the average of the conjugate functions $l_i^*(\cdot)$ as the prox function, namely $h(\cdot) = \frac{1}{n}\sum_{i=1}^n l_i^*(w_i)$, and it initializes the dual variable w^0 to be the prox-center (which is the point that minimizes the prox function). At the start of the ith iteration, the algorithm randomly chooses a coordinate j_i and computes the j_i^{th} coordinate of a subgradient of the dual objective function $D(w)$ at $w = w^i$, since indeed it is straightforward to verify that $\frac{1}{n}(X\tilde{\beta}^i - \nabla L^*(w^i))$ is a subgradient of $D(w)$ at $w = w^i$. The algorithm then performs a coordinate mirror descent step to update the dual variable w^i. Last of all – and optionally since it does not affect future computations – the algorithm updates the primal variable $\bar{\beta}^i$ in order to compute a primal-dual optimality gap certificate.

Algorithm 2 Randomized Coordinate Mirror Descent applied to the dual problem (10)

Initialize. Define the prox function $h(w) := \frac{1}{n}\sum_{i=1}^n l_i^*(w_i)$. Initialize with $w^0 = \arg\min_w \frac{1}{n}\sum_{i=1}^n l_i^*(w_i)$ and step-size sequences $\{\alpha_i\} \in (0,1]$ and $\{\eta_i\} \in (0,1]$. (Optional: set $\bar{\beta}^{-1} = 0$.)

For iterations $i = 0, 1, \ldots$
  Compute Randomized Coordinate of Subgradient of $D(\cdot)$ at w^i
    Compute $\tilde{\beta}^i \in \arg\min_\beta \left\{ \left(\frac{1}{n}(w^i)^T X\beta + R(\beta)\right) \right\}$

  Choose random index. Choose $j_i \in \mathcal{U}[1,\ldots,n]$
  Compute subgradient coordinate vector: $\tilde{g}^i \leftarrow \frac{1}{n}\left(x_{j_i}^T \tilde{\beta}^i - \dot{l}_{j_i}^*(w_{j_i}^i) \right) e_{j_i}$

  Update dual variable: Compute $w^{i+1} = \arg\min_w \left\{ \langle -\eta_i \tilde{g}^i, w - w^i \rangle + D_h(w, w^i) \right\}$

  (Optional Accounting:) $\bar{\beta}^i \leftarrow (1-\alpha_i)\bar{\beta}^{i-1} + \alpha_i \tilde{\beta}^i$.

The main result of this section is the following lemma concerning the equivalence of Algorithm 1 and Algorithm 2.

Lemma 2.1. (Equivalence Lemma) *Algorithm 1 and Algorithm 2 are equivalent as follows: the iterate sequence of either algorithm exactly corresponds to an iterate sequences of the other.* □

As a means to proving the lemma, we first reinterpret the update of w_{j_i} at iteration i of Algorithm 2 in the following proposition:

Proposition 2.1. *At iteration i of Algorithm 2 it holds that:*

(1.) $\dot{l}_{j_i}^*(w_{j_i}^{i+1}) = (1-\eta_i)\dot{l}_{j_i}^*(w_{j_i}^i) + \eta_i x_{j_i}^T \tilde{\beta}^i$, *and*

(2.) $w_{j_i}^{i+1} = \dot{l}_{j_i}\left((1-\eta_i)\dot{l}_{j_i}^*(w_{j_i}^i) + \eta_i x_{j_i}^T \tilde{\beta}^i \right)$.

Proof: Because $h(w)$ is a coordinate-wise separable function, we can rewrite the update for $w_{j_i}^{i+1}$

as

$$w_{j_i}^{i+1} = \arg\min_{w_{j_i}} \left\langle -\tfrac{\eta_i}{n}\left(x_{j_i}^T\tilde{\beta}^i - \dot{l}_{j_i}^*(w_{j_i}^i)\right), w_{j_i}\right\rangle + D_{\frac{1}{n}l_{j_i}^*}(w_{j_i}, w_{j_i}^i)$$

$$= \arg\min_{w_{j_i}} \left\langle -\eta_i\left(x_{j_i}^T\tilde{\beta}^i - \dot{l}_{j_i}^*(w_{j_i}^i)\right), w_{j_i}\right\rangle + D_{l_{j_i}^*}(w_{j_i}, w_{j_i}^i)$$

$$= \arg\min_{w_{j_i}} \left\langle -\eta_i x_{j_i}^T\tilde{\beta}^i - (1-\eta_i)\dot{l}_{j_i}^*(w_{j_i}^i), w_{j_i}\right\rangle + l_{j_i}^*(w_{j_i}) \ .$$

From the first-order optimality condition of the above 1-dimensional problem we have $\dot{l}_{j_i}^*(w_{j_i}^{i+1}) = \eta_i x_{j_i}^T\tilde{\beta}^i + (1-\eta_i)\dot{l}_{j_i}^*(w_{j_i}^i)$, which shows *(1.)*; and *(2.)* follows directly from *(1.)* by the properties of the conjugate function in Proposition A.1. □

Proof of Lemma 2.1 We show that the iterate sequence of Algorithm 2 corresponds exactly to an iterate sequence of Algorithm 1. The $\{s^i\}$ sequence is not formally defined in Algorithm 2, so let us define $s^i := \nabla L^*(w^i)$ for all $i = 0, 1, \ldots$, which is consistent through conjugacy with the relationship $w^i = \nabla L(s^i)$ in the Optional Accounting step of Algorithm 1 (see Proposition A.1). In order to show the correspondence we proceed by induction on the iteration counter i. For $i = 0$ we have from conjugacy that $s^0 := \nabla L^*(w^0) = 0$ from the definition w^0 in the initialization of Algorithm 2. We also need to show that $\tilde{\beta}^0$ is a solution to the linear optimization oracle problem in Algorithm 1. We have for all $i = 0, \ldots$, that:

$$\tilde{\beta}^i \in \arg\min_\beta \left\{ \tfrac{1}{n}\left(w^i\right)^T X\beta + R(\beta)\right\} = \arg\min_\beta \left\{ \tfrac{1}{n}\left(\nabla L(s^i)\right)^T X\beta + R(\beta)\right\}$$

$$= \arg\min_\beta \left\{ \left(d^i\right)^T\beta + R(\beta)\right\} \ ,$$

thus showing that β^i corresponds to a linear optimization oracle solution at iteration i in Algorithm 1 for all $i = 0, \ldots$. Now suppose that the correspondence holds for some iteration counter i, and let us examine $s^{i+1} := \nabla L^*(w^{i+1})$. We have from Proposition 2.1 that:

$$w_{j_i}^{i+1} = \dot{l}_{j_i}\left((1-\eta_i)\dot{l}_{j_i}^*(w_{j_i}^i) + \eta_i x_{j_i}^T\tilde{\beta}^i\right) = \dot{l}_{j_i}\left((1-\eta_i)s_{j_i}^i + \eta_i x_{j_i}^T\tilde{\beta}^i\right) \ , \tag{14}$$

where the first equality is from Proposition 2.1 and the second equality uses induction. This then implies that

$$(1-\eta_i)s_{j_i}^i + \eta_i x_{j_i}^T\tilde{\beta}^i = \dot{l}(w_{j_i}^{i+1}) = s^{i+1} \ .$$

And for all coefficient indices $j \neq i$ we have

$$s_j^{i+1} = \dot{l}^*(w_j^{i+1}) = \dot{l}(w_j^i) = s^i \ ,$$

where the second equality follows from conjugacy, whereby s^{i+1} satisfies the update rule as stated in Algorithm 1, thus demonstrating that the iterate sequence of Algorithm 2 corresponds exactly to an iterate sequence of Algorithm 1. The same type of analysis as above can be used to prove that the iterate sequence of Algorithm 1 corresponds exactly to an iterate sequence of Algorithm 2. □

3 Convergence Guarantees

In this section we develop computational guarantees for Algorithm 2, which automatically provide computational guarantees for Algorithm 1 due to the equivalence shown in Theorem 2.1. Our first – and main – result is Theorem 3.1, which is an expected $O(1/k)$ guaranteed decrease in the duality gap between (P) and (D). Secondly, in the case when $R(\cdot)$ is a strongly convex function, we present a linear convergence result on the duality gap in Theorem 3.2. We start by defining two measures – M and $D_{\max}$ – associated with (P) and whose values will enter our computational bounds.

Let $M := \max_{\beta \in \mathrm{dom}R(\cdot)} \max_{j=1,\ldots,n}\{|x_j^T \beta|\}$, and note that $M < +\infty$ since $\mathrm{dom}R(\cdot)$ is bounded by Assumption 1.1.

Let $\mathcal{W} \subset \mathbb{R}^n$ be the set of "optimal w responses" to values $\beta \in \mathrm{dom}R(\cdot)$ in the saddle-function $\phi(\beta, w)$, namely:

$$\mathcal{W} := \{\hat{w} \in \mathbb{R}^n : \hat{w} \in \arg\max_w \phi(\hat{\beta}, w) \text{ for some } \hat{\beta} \in \mathrm{dom}R(\cdot)\} \ ,$$

and let $D_{\max}$ be any upper bound on $D_h(\hat{w}, w^0)$ as $\hat{w}$ ranges over all values in $\mathcal{W}$, so that

$$D_h(\hat{w}, w^0) \leq D_{\max} \text{ for all } \hat{w} \in \mathcal{W} \ .$$

Note at the moment that there is no guarantee that $D_{\max} < +\infty$, but this will be remedied below in Proposition 3.1.

Proposition 3.1. *Under Assumption 1.1 it holds that $D_{\max} \leq \gamma M^2$.*

Before proving this proposition, we first show that there is a natural boundedness constraint for the dual problem:

Proposition 3.2. *Let $T := \{w \in \mathbb{R}^n : \|w - w^0\|_\infty \leq \gamma M\}$. Then:*

1. for any $\hat{\beta} \in \mathrm{dom}R(\cdot)$ it holds that $\arg\max_w \phi(\hat{\beta}, w) \in T$, and

2. for all w^i generated in Algorithm 2, it holds that $w^i \in T$.

Proof. We first prove *(1.)*. Notice that $w_0 = \nabla L(0)$ and $\arg\max_w \phi(\hat{\beta}, w) = \nabla L(X\hat{\beta})$ (from conjugacy via Proposition A.1) , whereby the γ-smoothness of $l_j(\cdot)$ implies that

$$\left\|\arg\max_w \phi(\hat{\beta}, w) - w_0\right\|_\infty = \left\|\nabla L(X\hat{\beta}) - \nabla L(0)\right\|_\infty = \max_j \left|\dot{l}_j(x_j^T\hat{\beta}) - \dot{l}_j(0)\right| \leq \gamma \max_j |x_j^T\hat{\beta}| \leq \gamma M \ ,$$

which proves *(1.)*. It follows from Proposition 3.1 that for any coordinate j and iterate i it holds that $\left|\dot{l}_j^*(w_j^i)\right| \leq M$. Together with $\dot{l}_j^*(w_j^0) = 0$, we have

$$\tfrac{1}{\gamma}\left|w_j^i - w_j^0\right| \leq \left|\dot{l}_j^*(w_j^i) - \dot{l}_j^*(w_j^0)\right| \leq M \ ,$$

(where the first inequality is from the $\frac{1}{\gamma}$-strong convexity of $l_j^*(w_j)$), from which it follows that $\|w^i - w^0\|_\infty \leq \gamma M$, which proves *(2.)*. $\square$

Proof of Proposition 3.1: Let $L(s) := \sum_{j=1}^n l_j(s_j)$ and $L^*(w) := \sum_{j=1}^n l_j^*(w_j)$, and note that $L(\cdot)$ and $L^*(\cdot)$ are a conjugate pair. Let $\hat{w} \in \mathcal{W}$ and let $\hat{\beta}$ be such that $\hat{w} \in \arg\max_w \phi(\hat{\beta}, w)$. Then

$$
\begin{aligned}
D_h(\hat{w}, w^0) &= \tfrac{1}{n} \left(L^*(\hat{w}) - L^*(w^0) \right) \\[2mm]
&= \tfrac{1}{n} \left((\hat{w})^T X \hat{\beta} - L(X\hat{\beta}) - L^*(w^0) \right) \\[2mm]
&\leq \tfrac{1}{n} \left(\max_{w \in T, \beta \in \mathrm{dom}R(\cdot)} \left\{ w^T X\beta - L(X\beta) \right\} - L^*(w^0) \right) \\[2mm]
&= \tfrac{1}{n} \left(\max_{w \in T, \beta \in \mathrm{dom}R(\cdot)} \left\{ (w - w^0)^T X\beta + \left(w^0\right)^T X\beta - L(X\beta) \right\} - L^*(w^0) \right) \\[2mm]
&\leq \tfrac{1}{n} \left(\max_{w \in T, \beta \in \mathrm{dom}R(\cdot)} \left\{ (w - w^0)^T X\beta \right\} + \max_{\beta \in \mathrm{dom}R(\cdot)} \left\{ \left(w^0\right)^T X\beta - L(X\beta) \right\} - L^*(w^0) \right) \\[2mm]
&\leq \tfrac{1}{n} \left(n \max_{w \in T, \beta \in \mathrm{dom}R(\cdot)} \| w - w^0 \|_\infty \| X\beta \|_\infty + L^*(w^0) - L^*(w^0) \right) \\[2mm]
&\leq \gamma M^2 \ ,
\end{aligned}
$$

where the second equality follows from Proposition A.1, the first inequality uses $\hat{\beta} \in \mathrm{dom}R(\cdot)$ and $\hat{w} \in T$ (from Proposition 3.2), and the last inequality uses $\max_{\beta \in \mathrm{dom}R(\cdot)} \| X\beta \|_\infty \leq M$. $\quad\square$

Remark 3.1. *A suitable value of $D_{\max}$ can often be easily derived based on the structure of $l_j(\cdot)$. For example, in logistic regression where the loss function is $l_j(s_j) := \log(1 + \exp(-y_j s_j))$ for the given label $y_j \in \{-1, 1\}$, we have $l_j^*(w_j) = -y_j w_j \ln(-y_j w_j) + (1 + y_j w_j) \ln(1 + y_j w_j)$ with $\mathrm{dom}l_j^*(\cdot) = \{w_j : 0 \leq -y_j w_j \leq 1\}$ (where $a \ln(a) := 0$ for $a = 0$). Therefore for all $\hat{w} \in \mathcal{W}$ it holds that*

$$
D_h(\hat{w}, w^0) \leq \max_{0 \leq -Yw \leq e} D_h(w, w^0) = \tfrac{1}{n} \left(\max_{0 \leq -Yw \leq e} L^*(w) - L^*(w^0) \right) = \ln(2) \ ,
$$

where Y is the diagonal matrix whose diagonal coefficients correspond to y and $e = [1, \ldots, 1]^T$, so we may set $D_{\max} = \ln(2)$.

Notice in Algorithm 1 and Algorithm 2 that j_i is a random variable; and that s^i, d^i, w^i, etc., are random variables that depend on all previous random variable values $j_0, j_1, \ldots, j_{i-1}$, and we denote this string of random variables by

$$
\xi_i = \{j_0, j_1, \ldots, j_{i-1}\} \ . \tag{15}
$$

We now state our main computational guarantee for Algorithm 2 (and hence for Algorithm 1 as well).

Theorem 3.1. *Consider the Stochastic Generalized Frank-Wolfe method (Algorithm 1) or the Randomized Dual Coordinate Mirror Descent method (Algorithm 2), with step-size sequences $\alpha_i = \frac{2(2n+i)}{(i+1)(4n+i)}$ and $\eta_i = \frac{2n}{2n+i+1}$ for $i = 0, 1, \ldots$. Denote*

$$
\bar{w}^k = \frac{2}{(4n+k)(k+1)} \sum_{i=0}^k (2n + i) w^i \ .
$$

13

Under Assumption 1.1, it holds for all $k \geq 0$ that

$$\mathbb{E}_{\xi_k}\left[P(\bar{\beta}^k) - D(\bar{w}^k)\right] \leq \frac{8n\gamma M^2}{(4n+k)} + \frac{2n(2n-1)D_{\max}}{(4n+k)(k+1)} \leq \frac{8n\gamma M^2}{(4n+k)} + \frac{2n(2n-1)\gamma M^2}{(4n+k)(k+1)} . \qquad \square$$

The following string of propositions will be needed for the proof of Theorem 3.1.

Proposition 3.3. *For all iterates i and any $j \in \{1, \ldots, n\}$ it holds that $\left| l_j^*(w_j^i) \right| \leq M$.*

Proof. We prove this by induction on i. The proposition is true for $i = 0$ because $l_j^*(w_j^0) = 0$ for all j by the definition of w^0. Next suppose that $\left| l_j^*(w_j^i) \right| \leq M$ for a given iterate i and for all $j = 1, \ldots, n$. Then at iteration $i + 1$ and any $j \neq j_i$ we have $w_j^{i+1} = w_j^i$, whereby $\left| l_j^*(w_j^{i+1}) \right| = \left| l_j^*(w_j^i) \right| \leq M$. And it follows from Proposition 2.1 that

$$\left| l_{j_i}^*(w_{j_i}^{i+1}) \right| = \left| (1 - \eta_i) \, l_{j_i}^*(w_{j_i}^i) + \eta_i x_{j_i}^T \tilde{\beta}^i \right| \leq (1 - \eta_i)M + \eta_i M = M ,$$

and therefore for any $j = 1, \ldots, n$, we have $\left| l_j^*(w_j^{i+1}) \right| \leq M$, which completes the proof by induction.
$\square$

As a simple corollary we obtain an upper bound on $\|\tilde{g}^i\|_2$ as follows:

Corollary 3.1. $\|\tilde{g}^i\|_2 = \frac{1}{n} \left| x_{j_i}^T \tilde{\beta}^i - l_{j_i}^*(w_{j_i}^i) \right| \leq \frac{2M}{n}.$

Proposition 3.4. $h(\cdot)$ *is $\frac{1}{n\gamma}$-strongly convex with respect to the norm $\| \cdot \|_2$.*

Proof. Recall that $h(w) = \frac{1}{n} \sum_{j=1}^n l_j^*(w_j)$. It follows from Assumption 1.1 and Proposition A.1 that $l_j^*(\cdot)$ is $\frac{1}{\gamma}$-strongly convex. Therefore for any $w^1, w^2 \in \text{dom}h(\cdot)$ it holds that:

$$
\begin{aligned}
h(w^1) &= \frac{1}{n} \sum_{j=1}^n l_j^*(w_j^1) \\[2mm]
&\geq \frac{1}{n} \sum_{j=1}^n \left(l_j^*(w_j^2) + l_j^{*\prime}(w_j^2)(w_j^1 - w_j^2) + \frac{1}{2\gamma}|w_j^2 - w_j^1|^2 \right) \\[2mm]
&= h(w^2) + \langle \nabla h(w^2), w^1 - w^2 \rangle + \frac{1}{2n\gamma}\|w^2 - w^1\|_2^2 . \qquad \square
\end{aligned}
$$

Proposition 3.5. $\phi(\tilde{\beta}^i, w) = D(w^i) + \left\langle \nabla_w \phi(\tilde{\beta}^i, w^i), w - w^i \right\rangle - D_h(w, w^i).$

Proof. The proof follows from straightforward substitution using $\phi(\tilde{\beta}^i, w) = \frac{1}{n}\left(w^T X \tilde{\beta}^i - \sum_{j=1}^n l_j^*(w_j) \right) + R(\tilde{\beta}^i)$ and noticing from the construction of $\tilde{\beta}^i$ that $D(w^i) = \phi(\tilde{\beta}^i, w^i)$.
$\square$

Proposition 3.6. *Consider the series $\{\alpha_i\}$ defined by $\alpha_i = \frac{2(2n+i)}{(4n+i)(i+1)}$ for $i \geq 0$ and define the series $\{\bar{\beta}^i\}$ by $\bar{\beta}^{-1} = 0$ and $\bar{\beta}^i = (1 - \alpha_i)\bar{\beta}^{i-1} + \alpha_i \tilde{\beta}^i$ for $i \geq 0$. Also define $\gamma_i = 2n + i$ for $i \geq 0$. Then*

$$\bar{\beta}^k = \frac{\sum_{i=0}^k \gamma_i \tilde{\beta}^i}{\sum_{i=0}^k \gamma_i} \quad \text{for all } k \geq 0 .$$

14

Proof: The proof follows easily by induction and using $\sum_{i=0}^{k} \gamma_i = \frac{(4n+k)(k+1)}{2}$. □

Proof of Theorem 3.1. Denote $g^i := \frac{1}{n}\left(X\tilde{\beta}^i - \nabla L^*(w^i)\right)$, whereby g^i is a subgradient of $D(w)$ at w^i, and $\tilde{g}^i$ is an unbiased estimator of g^i up to the scalar n, namely $\mathbb{E}_{j_i}[\tilde{g}^i] = \frac{1}{n}g^i$. Therefore we have for any i and any $w \in \mathcal{W}$ that:

$$
\begin{aligned}
\langle -g^i, w - w^i \rangle &= n\mathbb{E}_{j_i}\left[\langle -\tilde{g}^i, w - w^i \rangle\right] \\[2mm]
&\geq n\mathbb{E}_{j_i}\left[\langle -\tilde{g}^i, w^{i+1} - w^i \rangle + \tfrac{1}{\eta_i}D_h(w^{i+1}, w^i) + \tfrac{1}{\eta_i}D_h(w, w^{i+1}) - \tfrac{1}{\eta_i}D_h(w, w^i)\right] \\[2mm]
&\geq n\mathbb{E}_{j_i}\left[\langle -\tilde{g}^i, w^{i+1} - w^i \rangle + \tfrac{1}{2n\gamma\eta_i}\|w^{i+1} - w^i\|_2^2 + \tfrac{1}{\eta_i}D_h(w, w^{i+1}) - \tfrac{1}{\eta_i}D_h(w, w^i)\right] \\[2mm]
&\geq n\mathbb{E}_{j_i}\left[-\tfrac{1}{2}n\gamma\eta_i\|\tilde{g}^i\|_2^2 + \tfrac{1}{\eta_i}D_h(w, w^{i+1}) - \tfrac{1}{\eta_i}D_h(w, w^i)\right] \\[2mm]
&\geq -2\gamma M^2\eta_i + \tfrac{n}{\eta_i}\mathbb{E}_{j_i}[D_h(w, w^{i+1})] - \tfrac{n}{\eta_i}D_h(w, w^i) \,,
\end{aligned}
$$

(16)

where the first inequality is from the "three point property" of Tseng (Lemma A.1 in the Appendix), the second inequality is due to the fact that $h(w)$ is $\frac{1}{n\gamma}$-strongly convex with respect to the norm $\|\cdot\|_2$ (Proposition 3.4), and the third inequality is an application of the basic inequality $\langle x, y \rangle \leq \frac{1}{2}\|x\|_2^2 + \frac{1}{2}\|y\|_2^2$, and the last inequality uses Corollary 3.1.

On the other hand, we have from Proposition 3.5 that

$$\langle -g^i, w - w^i \rangle = \left\langle -\nabla_w \phi(\tilde{\beta}^i, w^i), w - w^i \right\rangle = D(w^i) - \phi(\tilde{\beta}^i, w) - D_h(w, w^i) \,. \tag{17}$$

Combining (16) and (17) and rearranging yields

$$-(\phi(\tilde{\beta}^i, w) - D(w^i)) \geq -2\gamma M^2\eta_i + \tfrac{n}{\eta_i}\mathbb{E}_{j_i}[D_h(w, w^{i+1})] - \left(\tfrac{n}{\eta_i} - 1\right)D_h(w, w^i).$$

Substituting $\eta_i = \frac{2n}{2n+i+1}$ and multiplying by $2n+i$ results, we arrive at the following inequality after rearranging terms:

$$
\begin{aligned}
&(2n+i)(\phi(\tilde{\beta}^i, w) - D(w^i)) \\
&\leq\ 4n\gamma M^2\left(\tfrac{2n+i}{2n+i+1}\right) + \tfrac{1}{2}\left((2n+i)(2n+i-1)D_h(w, w^i) - (2n+i)(2n+i+1)\mathbb{E}_{j_i}[D_h(w, w^{i+1})]\right)\,.
\end{aligned}
$$

Summing the above inequality for $i = 0, \ldots, k$ and recalling from Proposition 3.6 that $\bar{\beta}^k := \frac{2}{(4n+k)(k+1)}\sum_{i=0}^{k}(2n+i)\tilde{\beta}^i$, and taking the unconditional expectation over ξ_k (recall the definition of ξ_k in (15)), we arrive at:

$$
\begin{aligned}
\tfrac{(4n+k)(k+1)}{2}\mathbb{E}_{\xi_k}[\phi(\bar{\beta}^k, w) - D(\bar{w}^k)] &= \left(\sum_{i=0}^{k} 2n+i\right)\mathbb{E}_{\xi_k}[\phi(\bar{\beta}^k, w) - D(\bar{w}^k)] \\[2mm]
&\leq\ \mathbb{E}_{\xi_k}\left[\sum_{i=0}^{k}(2n+i)(\phi(\tilde{\beta}^i, w) - D(w^i))\right] \\[2mm]
&\leq\ 4(k+1)n\gamma M^2 + \tfrac{1}{2}(2n)(2n-1)D_h(w, w^0) \\[2mm]
&\leq\ 4(k+1)n\gamma M^2 + n(2n-1)D_{\max} \,,
\end{aligned}
$$

15

where the first inequality uses the convexity of $\phi(\beta, w)$ over β and the concavity of $D(w)$, the second inequality follows from the summation and canceling terms in the telescoping series, and the third inequality uses $w \in \mathcal{W}$. Choosing $\hat{w} = \arg\max_w \phi(\bar{\beta}^k, w)$, we have $P(\bar{\beta}^k) = \phi(\bar{\beta}^k, \hat{w})$, which yields:

$$\mathbb{E}_{\xi_k}[P(\bar{\beta}^k) - D(\bar{w}^k)] \leq \frac{8n\gamma M^2}{(4n+k)} + \frac{2n(2n-1)D_{\max}}{(4n+k)(k+1)} ,$$

thus showing the first inequality in the statement of the theorem. The second inequality in the statement of the theorem then follows as a simple application of Proposition 3.1. $\qquad\square$

3.1 Linear Convergence when $R(\cdot)$ is Strongly Convex

In this section, we further assume $R(\cdot)$ is a μ-strongly convex function, and we develop a linear convergence guarantee for Algorithms 1 and 2 . We first formally define a separable function.

Definition 3.1. *The function $h : R^n \to R$ is separable if*

$$h(x) = \sum_{i=1}^{n} h_i(x_i),$$

where x_i is the ith coordinate of x and h_i is a univariate function.

Next we introduce the notation of relative smoothness and relative strong convexity developed recently in [25][24][5][14][44]. We adapt a simplified version of the coordinate-wise relative smoothness condition as in [14].

Definition 3.2. *$f(\cdot)$ is coordinate-wise σ-smooth relative to a separable reference function $h(\cdot)$ if for any x, scalar t and coordinate j it holds that:*

$$f(x + te_j) \leq f(x) + \langle \nabla f(x), te_j \rangle + \sigma D_h(x + te_j, x) . \tag{18}$$

We also adapt the notion of relative strong convexity developed in [25].

Definition 3.3. *$f(\cdot)$ is μ-strongly convex relative to $h(\cdot)$ if for any x, y, it holds that*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \mu D_h(y, x) . \tag{19}$$

The next proposition states that the dual function $D(w)$ is both coordinate-wise smooth and strongly concave relative to the reference function $h(w) := \frac{1}{n} \sum_{j=1}^{n} l_j^*(w_j)$. In the proposition, recall that x_j is the jth row of the matrix X.

Proposition 3.7.

(1.) Suppose $R(\cdot)$ is a μ-strongly convex function with respect to $\|\cdot\|_2$, then $-D(\cdot)$ is $coordinate-$ $wise \left(\frac{\gamma \max_j \|x_j\|_2^2}{n\mu} + 1\right)$-smooth relative to $h(\cdot)$, and

(2.) $-D(\cdot)$ is $1-$strongly convex relative to $h(\cdot)$.

Proof. *(1.)* Consider w_1 and w_2 such that $w_2 = w_1 + te_j$ for some coordinate j, namely w_1 and w_2 only differ in one coordinate. It follows from Proposition A.1 that $R^*(\cdot)$ is $\frac{1}{\mu}$-smooth with respect to $\|\cdot\|_2$, thus we have

$$
\begin{aligned}
R^*\left(-\tfrac{1}{n}X^T w_2\right) &\leq R^*\left(-\tfrac{1}{n}X^T w_1\right) + \left\langle \nabla R^*\left(-\tfrac{1}{n}X^T w_1\right), -\tfrac{1}{n}X^T(w_2 - w_1)\right\rangle + \tfrac{1}{2\mu}\left\|\tfrac{1}{n}X^T(w_2 - w_1)\right\|_2^2 \\[2mm]
&= R^*\left(-\tfrac{1}{n}X^T w_1\right) + \left\langle -\tfrac{1}{n}X\nabla R^*\left(-\tfrac{1}{n}X^T w_1\right), w_2 - w_1\right\rangle + \tfrac{t^2}{2n^2\mu}\|x_j\|_2^2 \\[2mm]
&\leq R^*\left(-\tfrac{1}{n}X^T w_1\right) + \left\langle -\tfrac{1}{n}X\nabla R^*\left(-\tfrac{1}{n}X^T w_1\right), w_2 - w_1\right\rangle + \tfrac{\gamma\|x_j\|_2^2}{n\mu}D_h(w_2, w_1),
\end{aligned}
$$

where the first inequality follows from smoothness, the equality is from $w_2 = w_1 + te_j$, and the last inequality utilizes the fact that $h(\cdot)$ is $(\frac{1}{n\gamma})$-strongly convexity with respect to $\|\cdot\|_2$. Therefore it holds that $\hat{f}(w) := R^*\left(-\frac{1}{n}X^T w\right)$ is coordinate-wise $(\frac{\gamma\max_j\|x_j\|_2^2}{n\mu})$-smooth relative to $h(\cdot)$. The proof is completed by noticing that $-D(w) = R^*\left(-\frac{1}{n}X^T w\right) + h(w)$.

(2.) This follows from the additivity property of relative strong convexity (Proposition 1.2 in [25]), whereby $D(\cdot)$ is 1-strongly concave relative to $h(w)$. □

The following theorem states a linear convergence guarantee in the case when $R(\cdot)$ is strongly convex.

Theorem 3.2. *Suppose $D(\cdot)$ is coordinate-wise σ-smooth relative to $h(\cdot)$. Consider the Stochastic Generalized Frank-Wolfe method (Algorithm 1) or the Randomized Dual Coordinate Mirror Descent method (Algorithm 2), with step-size sequences $\eta_i = \frac{1}{\sigma}$ and $\alpha_i = \frac{n^{-1}\sigma^i}{\sigma^{i+1} - (\sigma - 1/n)^{i+1}}$. Under Assumption 1.1 it holds for all $k \geq 1$ that*

$$
\mathbb{E}_{\xi_k}\left[P(\bar{\beta}^{k-1}) - D(w^k)\right] \leq \frac{D_{\max}}{\left(1 + \frac{1}{n\sigma - 1}\right)^k - 1} \leq \frac{\gamma M^2}{\left(1 + \frac{1}{n\sigma - 1}\right)^k - 1} . \tag{20}
$$

□

Notice that the first inequality in (20) shows linear convergence; indeed, in this case it holds that

$$
\frac{1}{\left(1 + \frac{1}{n\sigma - 1}\right)^k - 1} \leq n\sigma\left(1 - \frac{1}{n\sigma}\right)^k . \tag{21}
$$

(This inequality holds trivially for $k = 1$, and induction on k establishes the result for $k \geq 2$.) Furthermore, when k is large the -1 term in the denominator of the left-hand side can be ignored which yields the asymptotic bound $\left(1 - \frac{1}{n\sigma}\right)^k D_{\max}$. The next corollary states the implication of this linear convergence bound in terms of the values γ and μ of the γ-smoothness of $l_1(\cdot), \ldots, l_n(\cdot)$ and the μ-strong convexity of $R(\cdot)$.

Corollary 3.2. *Choose* $\sigma = \frac{\gamma \max_j \|x_j\|_2^2}{n\mu} + 1$ *as per Proposition 3.7. Then Theorem 3.2 and* (21) *imply*

$$\mathbb{E}_{\xi_k}\left[P(\bar{\beta}^{k-1}) - D(w^k)\right] \leq \frac{D_{\max}}{\left(1 + \frac{1}{\frac{\gamma \max_j \|x_j\|_2^2}{\mu} + n - 1}\right)^k - 1} \leq D_{\max}\left(\frac{\gamma \max_j \|x_j\|_2^2}{\mu} + n\right)\left(1 - \frac{1}{n + \frac{\gamma \max_j \|x_j\|_2^2}{\mu}}\right)^k.$$

Before proving Theorem 3.2, we first present an elementary proposition for a separable reference function $h(\cdot)$, whose proof is given in Appendix A.3.

Proposition 3.8. *Suppose* $h(\cdot) : \mathbb{R}^n \to \mathbb{R}$ *is a separable function. Let* $j \sim \mathcal{U}[1, \ldots, n]$. *For given* $x, a, y \in \mathbb{R}^n$, *define the random variable* $b \in R^n$ *such that* $b_j = a_j$, *and* $b_i = x_i$ *for all* $i \neq j$. *Then:*

$$D_h(y, a) - D_h(y, x) = n\mathbb{E}_j\left(D_h(y, b) - D_h(y, x)\right). \quad \square$$

We also will use the following proposition whose proof follows easily by induction on k.

Proposition 3.9. *Consider the series* $\{\alpha_i\}$ *defined by* $\alpha_i = \frac{n^{-1}\sigma^i}{\sigma^{i+1} - (\sigma - 1/n)^{i+1}}$ *for* $i \geq 0$, *and define the series* $\{\bar{\beta}^i\}$ *by* $\bar{\beta}^{-1} = 0$ *and* $\bar{\beta}^i = (1 - \alpha_i)\bar{\beta}^{i-1} + \alpha_i\tilde{\beta}^i$ *for* $i \geq 0$. *Also define* $\gamma_i = \left(\frac{n\sigma}{n\sigma - 1}\right)^i$ *for* $i \geq 0$. *Then*

$$\bar{\beta}^k = \frac{\sum_{i=0}^k \gamma_i \tilde{\beta}^i}{\sum_{i=0}^k \gamma_i} \quad \text{for all } k \geq 0 .$$

$\square$

Proof of Theorem 3.2.

Notice that $\tilde{g}^i = \nabla_{j_i} D(w^i)e_{j_i}$, and w^{i+1} is a coordinate update from w^i, whereby we have

$$-D(w^{i+1}) \leq -D(w^i) - \langle \tilde{g}^i, w^{i+1} - w^i\rangle + \sigma D_h(w^{i+1}, w^i) \leq -D(w^i) ,$$

and hence the dual function value sequence $\{D(w^i)\}$ is non-decreasing.

Define $r^{i+1} := \arg\min_w \{\langle -\nabla D(w^i), w - w^i\rangle + \sigma D_h(w, w^i)\}$, then we have

$$
\begin{aligned}
\mathbb{E}_{j_i}[-D(w^{i+1})] \ &\leq \mathbb{E}_{j_i}[-D(w^i) - \langle \nabla D(w^i), w^{i+1} - w^i\rangle + \sigma D_h(w^{i+1}, w^i)] \\
&= \mathbb{E}_{j_i}[-D(w^i) - \tfrac{1}{n}\left(\langle \nabla D(w^i), r^{i+1} - w^i\rangle + \sigma D_h(r^{i+1}, w^i)\right)] \\
&\leq \mathbb{E}_{j_i}[-D(w^i) - \tfrac{1}{n}\left(\langle \nabla D(w^i), w - w^i\rangle + \sigma D_h(w, w^i) - \sigma D_h(w, r^{i+1})\right)] \\
&= \mathbb{E}_{j_i}[-D(w^i) - \tfrac{1}{n}\langle \nabla D(w^i), w - w^i\rangle + \sigma D_h(w, w^i) - \sigma D_h(w, w^{i+1})] \\
&= \mathbb{E}_{j_i}[-\tfrac{n-1}{n}D(w^i) - \tfrac{1}{n}\left(D(w^i) + \langle \nabla D(w^i), w - w^i\rangle\right) + \sigma D_h(w, w^i) - \sigma D_h(w, w^{i+1})] ,
\end{aligned}
$$

(22)

where the first inequality is from the coordinate-wise σ-smoothness of $D(w)$ relative to $h(w)$ and the fact that w^{i+1} is a coordinate update from w^i, the first equality is due to expectation and the separability of $h(\cdot)$, the second inequality uses the three-point property (Lemma A.1), the second equality uses Proposition 3.8, and the third equality is just arithmetic rearrangement.

Notice that

$$-D(w^i) - \left\langle \nabla D(w^i), w - w^i \right\rangle + n\sigma D_h(w, w^i) - n\sigma D_h(w, w^{i+1})$$

$$= -D(w^i) - \left\langle \nabla_w \phi(\tilde{\beta}^i, w^i), w - w^i \right\rangle + n\sigma D_h(w, w^i) - n\sigma D_h(w, w^{i+1})$$

$$= -\phi(\tilde{\beta}^i, w) + (n\sigma - 1)D_h(w, w^i) - n\sigma D_h(w, w^{i+1}) \ ,$$

where the last equality utilizes Proposition 3.5. We can then rewrite (22) (after multiplying by n on both sides) as

$$\mathbb{E}_{j_i}[-nD(w^{i+1})] \le \mathbb{E}_{j_i}\left[-(n-1)D(w^i) - \phi(\tilde{\beta}^i, w) + (n\sigma - 1)D_h(w, w^i) - n\sigma D_h(w, w^{i+1})\right] \ . \quad (23)$$

Multiplying (23) by $\left(\frac{n\sigma}{n\sigma-1}\right)^{i+1}$ and summing over $i = 0, \ldots, k-1$, we obtain:

$$\mathbb{E}_{\xi_k}\left[-\sum_{i=1}^{k} n\left(\frac{n\sigma}{n\sigma-1}\right)^i D(w^i)\right]$$

$$\le \ \mathbb{E}_{\xi_k}\left[-\sum_{i=1}^{k}(n-1)\left(\frac{n\sigma}{n\sigma-1}\right)^i D(w^{i-1}) - \sum_{i=1}^{k}\left(\frac{n\sigma}{n\sigma-1}\right)^i \phi(\tilde{\beta}^{i-1}, w) + n\sigma D_h(w, w^0)\right]$$

$$\le \ \mathbb{E}_{\xi_k}\left[-\sum_{i=1}^{k}(n-1)\left(\frac{n\sigma}{n\sigma-1}\right)^i D(w^{i-1}) - \left(\sum_{i=1}^{k}\left(\frac{n\sigma}{n\sigma-1}\right)^i\right) \phi(\bar{\beta}^{k-1}, w) + n\sigma D_h(w, w^0)\right] \ ,$$

where the last inequality is from Proposition 3.9 and the convexity of $\phi(\beta, w)$ in β. Since the sequence $\{D(w^i)\}$ is non-decreasing in i it follows that:

$$\mathbb{E}_{\xi_k}\left[-\left(\sum_{i=1}^{k}\left(\frac{n\sigma}{n\sigma-1}\right)^i\right) D(w^k)\right] \le \mathbb{E}_{\xi_k}\left[-\left(\sum_{i=1}^{k}\left(\frac{n\sigma}{n\sigma-1}\right)^i\right) \phi(\bar{\beta}^{k-1}, w)\right] + n\sigma D_h(w, w^0) \ . \quad (24)$$

Let us substitute the following value of w in (24): $w \leftarrow \hat{w}^{k-1} := \arg\max_w\{\phi(\bar{\beta}^{k-1}, w)\}$, which yields:

$$\left(\sum_{i=1}^{k}\left(\frac{n\sigma}{n\sigma-1}\right)^i\right) \mathbb{E}_{\xi_k}\left[\phi(\bar{\beta}^{k-1}, \hat{w}^{k-1}) - D(w^k)\right] \le n\sigma D_h(\hat{w}^{k-1}, w^0) \le n\sigma D_{\max} \ ,$$

where the last inequality above comes from the definition of $D_{\max}$. Therefore we have

$$\mathbb{E}_{\xi_k}\left[P(\bar{\beta}^{k-1}) - D(w^k)\right] \le \frac{n\sigma}{\left(\sum_{i=1}^{k}\left(\frac{n\sigma}{n\sigma-1}\right)^i\right)} D_{\max} = \frac{D_{\max}}{\left(1 + \frac{1}{n\sigma-1}\right)^k - 1} \ ,$$

which furnishes the proof by utilizing Proposition 3.9. $\qquad \square$

Remark 3.2. *Algorithm 1 as well as the convergence analysis in Theorem 3.1 and Theorem 3.2 can be directly extended to the mini-batch setting. The algorithm extension is accomplished by replacing the single randomly chosen index j_i in the statement of Algorithm 1 by a random subset of the indices; and the analysis of the algorithm needs to then use the mini-batch of samples instead of a single sample. More specifically, in the mini-batch version of Algorithm 1, we pre-set the batch size b, and at iteration i we choose a random subset $B_i \subseteq \{1, 2, \ldots, n\}$ of the indices uniformly without replacement such that $|B_i| = b$ at the i-th iteration. We then update the predicted values s using only the indices in the subset B_i, namely $s_j^{i+1} = (1 - \eta_i)s_j^i + \eta_i(x_j^T \tilde{\beta}^i)$ for $j \in B_i$, and $s_j^{i+1} = s_j^i$ for $j \notin B_i$. The update of the substitute gradient becomes $d^{i+1} = \frac{1}{n}X^T \nabla L(s^{i+1}) = d^i + \frac{1}{n}\sum_{j \in B_i} \left(\dot{l}_j(s_j^{i+1}) - \dot{l}_j(s_j^i) \right) x_j$. The rest of the algorithm remains the same as stated in Algorithm 1.*

By a similar analysis to that of Theorem 3.1 and Theorem 3.2, we can obtain similar convergence guarantees for the above mini-batch version of SGFW. Essentially we just need to replace n by n/b in the statement of Theorem 3.1 and Theorem 3.2 as well as most of the places in the analysis (except the third and fourth inequalities in (16)), after noticing that Corollary 3.1 becomes $\|\tilde{g}^i\|_2 \leq \frac{2M\sqrt{b}}{n}$. In particular, for the non-strongly convex case, the convergence guarantee becomes

$$\mathbb{E}_{\xi_k}\left[P(\bar{\beta}^k) - D(\bar{w}^k) \right] \leq \frac{8(\frac{n}{b})\gamma M^2}{\left(4(\frac{n}{b}) + k\right)} + \frac{2(\frac{n}{b})(2(\frac{n}{b}) - 1)\gamma M^2}{(4(\frac{n}{b}) + k)(k+1)} \ ,$$

and for the strongly-convex case, the convergence guarantee becomes

$$\mathbb{E}_{\xi_k}\left[P(\bar{\beta}^{k-1}) - D(w^k) \right] \leq \frac{D_{\max}}{\left(1 + \frac{1}{\sigma(\frac{n}{b})-1}\right)^k - 1} \leq \frac{\gamma M^2}{\left(1 + \frac{1}{\sigma(\frac{n}{b})-1}\right)^k - 1} \ .$$

Moreover, the updates of Algorithm 1 in the mini-batch setting can be implemented in parallel as a result of the separability of samples in Algorithm 1.

Remark 3.3. *A natural question to ask next is whether one can achieve an accelerated convergence rate when $R(\cdot)$ is strongly convex, similar to that in [40], [23]. The answer actually is yes, as one can utilize similar proof techniques as those developed in [23]. However, the accelerated version may not have a natural interpretation in the primal variables.*

4 Computational Experiments and Comparisons

In this section we present the results of some basic numerical experiments where we compare Algorithm 1 (SGFW) with the following four other stochastic Frank-Wolfe methods in the recent literature:

- SCGS – stochastic gradient sliding algorithm proposed in [22];

- SFW – stochastic Frank-Wolfe algorithm proposed in [17];

- SVRF – stochastic variance reduction Frank-Wolfe algorithm proposed in [17]; and

dataset	sample size (n)	feature size (p)
a9a	32561	123
w8a	49749	300
mushrooms	8124	112
gisette	6000	5000

Table 1: Dimensions and value of δ for four LIBSVM data instances.

Batch Size	SGFW	SFW	SCGM	SCGS	SVRF
Size consistent with theory	$0.01n$	$O(k^2)$	$0.01n$	$O(k^3)$	$O(k)$
Size used in practice	$0.01n$	$\min\{k^2, n/2\}$	$0.01n$	$\min\{k^3, n/2\}$	$\min\{k, n/2\}$

Table 2: Batch sizes used in the five stochastic Frank-Wolfe methods implemented in our computational experiments.

- SCGM – stochastic conditional gradient method proposed in [30].

We analyzed the performance of these five algorithms on instances of the following ℓ_1 norm constrained sparse logistic regression problem:

$$\min_{\beta \in \mathbb{R}^p} \quad P(\beta) = \frac{1}{n} \sum_{j=1}^{n} \ln(1 + \exp(-y_j x_j^T \beta))$$

$$\text{s.t.} \quad \|\beta\|_1 \leq \delta \ . \tag{25}$$

We ran the five stochastic Frank-Wolfe algorithms on ten dataset instances of the constrained logistic regression problem (25) in LIBSVM [8]. Here we report on four of these dataset instances, namely a9a, w8a, mushrooms, and gisette, as the results on these four datasets are typical of the results of the other datasets. Table 1 describes the dimensions for these four data instances. We set $\delta = 5$ for our experiments with these four datasets, which resulted in solutions on the boundary of the feasible region in all instances.

Instead of using one single sample (batch-size equal to 1) per iteration, we found it far more efficient to run SGFW and SCGM using a mini-batch. So as not to over-engineer our analysis or unduly bias our results, we used a mini-batch size of 1% of the training data size ($.01n$) and chose all batches randomly without replacement at all iterations (see Remark 3.2 for a discussion of how to modify SGFW (Algorithm 1) using mini-batches, with associated modifications of the computational guarantees). Note that in theory, algorithms SCGS, SFW, and SVRF all require increasing the batch size to $O(k^3)$, $O(k^2)$, and $O(k)$ at iteration k, respectively. In order to retain actual stochasticity of these three methods, we set the maximum batch-size to be 50% of all samples for these three methods; thus when the batch size specified in each of these algorithms is larger than $n/2$, we randomly select $n/2$ samples without replacement in constructing the stochastic gradient estimator. Table 2 summarizes the above discussion of batch-size modifications for our computational experiments.
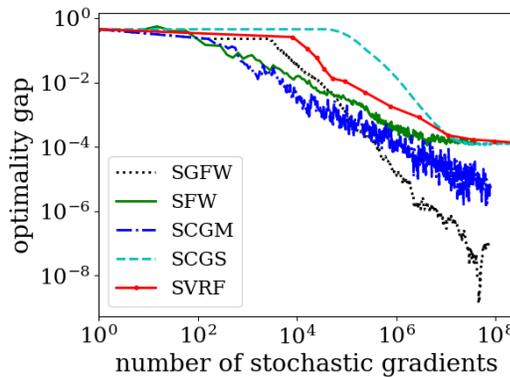
Figure 1 shows the optimality gap versus the number of stochastic gradient computations (counting one for each sample in each batch) for the five different stochastic Frank-Wolfe methods. In each sub-figure, the vertical axis is the objective value optimality gap $P(\beta^k) - P(\beta^*)$ in log scale, where $P(\beta^*)$ is estimated after-the-fact using the best solution obtained over the iterations; the horizontal
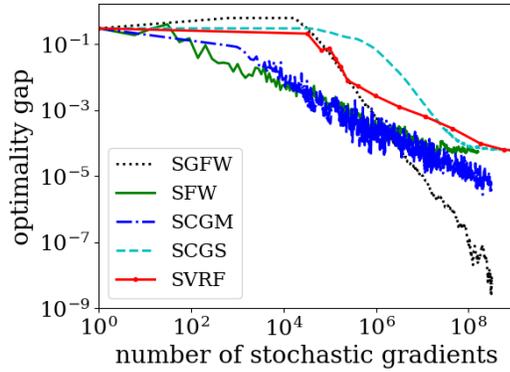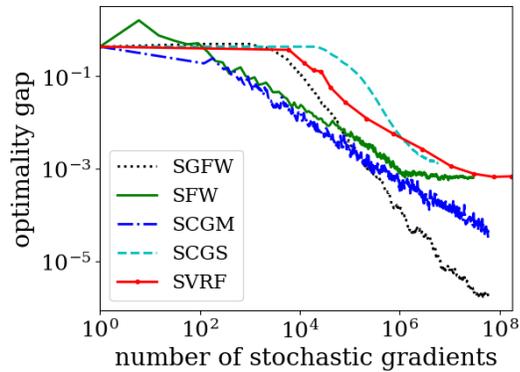
(a) a9a dataset
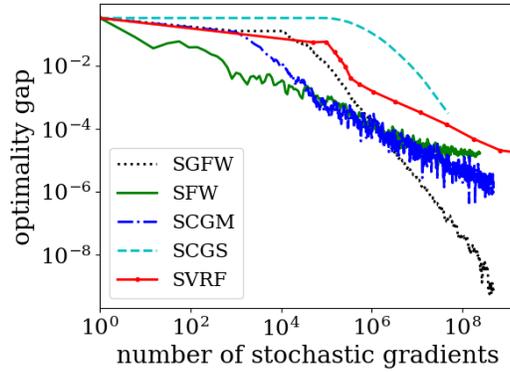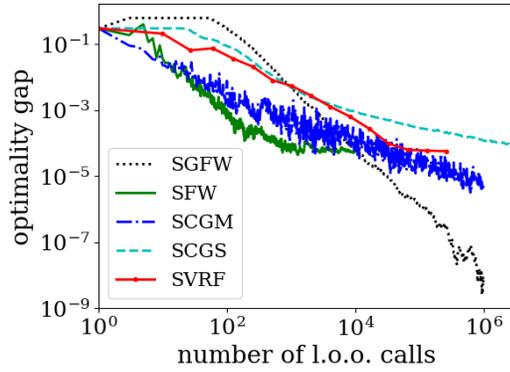
(b) w8a dataset

(c) mushrooms dataset

(d) gisette dataset

Figure 1: Figure showing the optimality gap versus the number of stochastic gradient computations (counting one for each sample in each batch) for each of the five different stochastic Frank-Wolfe methods, for the a9a, w8a, mushrooms, and gisette dataset instances.

axis in each sub-figure is the number of stochastic gradient computations computed so far, also in log scale. Here we see that although the optimality gap of SFW and SCGM may have faster decay initially, nevertheless GSFW decays much faster than the other stochastic Frank-Wolfe variants after a while. This is consistent with the theory that SGFW has a superior order of convergence rate ($O(1/\varepsilon)$ stochastic gradient computations) for solving (25).

Figure 2 shows the optimality gap versus the number of linear optimization oracle calls (solving the linear optimization subproblem) for the five different stochastic Frank-Wolfe methods. In each sub-figure, the vertical axis is the objective value optimality gap in log scale, and the horizontal axis is the number of linear optimization oracle calls, also in log scale. Similar to Figure 1, here we see again that the optimality gap of GSFW decays much faster than the other stochastic Frank-Wolfe methods after a while, which is again consistent the superior order of convergence rate of GSFW ($O(1/\varepsilon)$ linear optimization oracle calls) for solving (25).
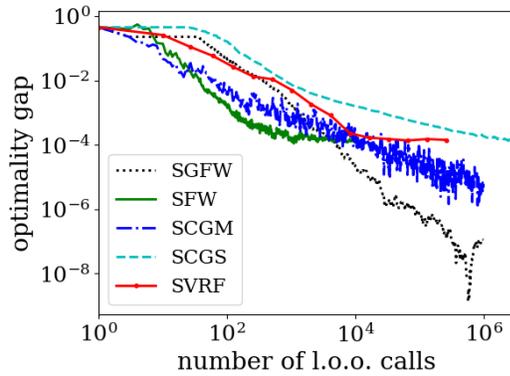
Table 3 shows the number of stochastic gradient computations and the number of linear optimiza-
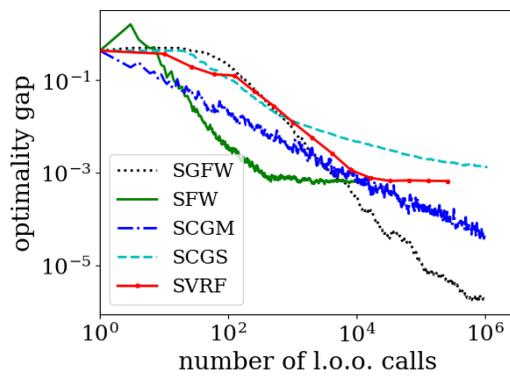
(a) a9a dataset
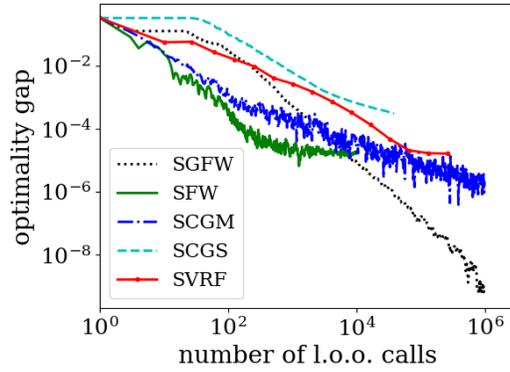
(b) w8a dataset

(c) mushrooms dataset

(d) gisette dataset

Figure 2: Figure showing the optimality gap versus the number of linear optimization oracle calls for each of the five different stochastic Frank-Wolfe methods, for the a9a, w8a, mushrooms, and gisette dataset instances.

tion oracle calls to achieve an accuracy of 10^{-5} for the four LIBSVM datasets evaluated in detail herein, for the (deterministic) Frank-Wolfe method, SGFW, and SCGM. (SCGS, SFW, and SVRF do not achieve an accuracy of 10^{-5} in a reasonable number of iterations, so we do not present results for these three methods in Table 3.) Table 3 shows that SGFW usually requires fewer gradient computations, while the deterministic Frank-Wolfe method clearly dominates the stochastic Frank-Wolfe methods in term of the number of linear optimization oracle calls. Of course, the Frank-Wolfe method utilizes the exact gradient at each linear optimization oracle call, while the stochastic Frank-Wolfe methods utilize an inexact gradient at each such linear optimization oracle call, and this is likely the reason why the deterministic Frank-Wolfe method requires dramatically fewer linear optimization oracle calls to achieve the given desired optimality gap accuracy.

Dataset	Frank-Wolfe		SGFW		SCGM	
	total sample gradient calls ($\times 10^6$)	linear optimization oracle calls	Stochastic gradient calls ($\times 10^6$)	linear optimization oracle calls	Stochastic Gradient calls ($\times 10^6$)	linear optimization oracle calls
a9a	14.5	**448**	**10.3**	31,900	65.3	201,000
w8a	9.65	**194**	**4.21**	8,470	6.92	13,900
mushroom	6.44	**793**	**1.27**	15,700	7.4	91,400
gisette	**6.27**	1045	6.56	109,000	–	–

Table 3: Comparison of the number of stochastic gradient computations and the number of linear optimization oracle calls, to achieve an accuracy of 10^{-5} for solving (25) for the (deterministic) Frank-Wolfe method, SGFW, and SCGM, for the four LIBSVM datasets a9a, w8a, mushrooms, and gisette.

Appendix

A.1 Properties of Conjugate Functions

Recall the definition of the conjugate of a function $f(\cdot)$:

$$f^*(y) := \sup_{x \in \text{dom } f} \{y^T x - f(x)\} \ .$$

The following properties of conjugate functions are used in this paper:

Proposition A.1. *(see [3], [46], [21]) If $f(\cdot)$ is a closed convex function, then $f^{**}(\cdot) = f(\cdot)$. Furthermore:*

1. *$f(\cdot)$ is γ-smooth with domain $\mathbb{R}^p$ with respect to the norm $\| \cdot \|$ if and only if $f^*(\cdot)$ is $1/\gamma$-strongly convex with respect to the (dual) norm $\| \cdot \|^*$.*

2. *If $f(\cdot)$ is differentiable and strictly convex, then the following three conditions are equivalent:*

   (a) $y = \nabla f(x)$

   (b) $x = \nabla f^*(y)$, and

   (c) $x^T y = f(x) + f^*(y)$.

A.2 Three-Point Property

We state here the "three-point property" as memorialized by Tseng [43]:

Lemma A.1. (Three-Point Property [43]) *Let $\phi(x)$ be a convex function, and let $D_h(\cdot, \cdot)$ be the Bregman distance for $h(\cdot)$. For a given vector z, let*

$$z^+ := \arg \min_{x \in Q} \{\phi(x) + D_h(x, z)\} \ .$$

Then

$$\phi(x) + D_h(x, z) \geq \phi(z^+) + D_h(z^+, z) + D_h(x, z^+) \quad \text{for all } x \in Q \ . \quad \square$$

A.3 Proof of Proposition 3.8

Note that

$$
\begin{aligned}
\langle \nabla h\left(a\right) - \nabla h\left(x\right), y\rangle &= \sum_{i=1}^{n} \langle \nabla h_i\left(a_i\right) - \nabla h_i\left(x_i\right), y_i\rangle \\
&= n\mathbb{E}_j \langle \nabla h_j\left(a_j\right) - \nabla h_j\left(x_j\right), y_j\rangle \\
&= n\mathbb{E}_j \langle \nabla h_j\left(b_j\right) - \nabla h_j\left(x_j\right), y_j\rangle \\
&= n\mathbb{E}_j \sum_{i=1}^{n} \langle \nabla h_i\left(b_i\right) - \nabla h_i\left(x_i\right), y_i\rangle \\
&= n\mathbb{E}_j \langle \nabla h\left(b\right) - \nabla h\left(x\right), y\rangle \;,
\end{aligned}
$$

where the second equation is from expectation, and the third and fourth equation follow because $b_j = a_j$ and $b_i = x_i$ for all $i \neq j$. Using similar logic it also holds that

$$
\langle \nabla h\left(a\right), a\rangle - \langle \nabla h\left(x\right), x\rangle \;=\; n\mathbb{E}_j \left(\langle \nabla h\left(b\right), b\rangle - \langle \nabla h\left(x\right), x\rangle\right) \;,
$$

and

$$
h\left(a\right) - h\left(x\right) \;=\; n\mathbb{E}_j \left(h\left(b\right) - h\left(x\right)\right).
$$

Therefore,

$$
\begin{aligned}
D_h\left(y, a\right) - D_h\left(y, x\right) &= \langle \nabla h\left(a\right), y - a\rangle - \langle \nabla h\left(x\right), y - x\rangle - \left(h\left(a\right) - h\left(x\right)\right) \\
&= \langle \nabla h\left(a\right) - \nabla h\left(x\right), y\rangle - \left(\langle \nabla h\left(a\right), a\rangle - \langle \nabla h\left(x\right), x\rangle\right) - \left(h\left(a\right) - h\left(x\right)\right) \\
&= n\mathbb{E}_j \left[\langle \nabla h\left(b\right) - \nabla h\left(x\right), y\rangle - \left(\langle \nabla h\left(b\right), b\rangle - \langle \nabla h\left(x\right), x\rangle\right) - \left(h\left(b\right) - h\left(x\right)\right)\right] \\
&= n\mathbb{E}_j \left[\langle \nabla h\left(b\right), y - b\rangle - \langle \nabla h\left(x\right), y - x\rangle - \left(h\left(b\right) - h\left(x\right)\right)\right] \\
&= n\mathbb{E}_j \left[D_h\left(y, b\right) - D_h\left(y, x\right)\right] \;. \quad \square
\end{aligned}
$$

A.4 Connections and Comparisons between GSFW and Stochastic Dual Coordinate Ascent Methods

In this subsection we discuss connections and comparisons between GSFW (Algorithm 1, and equivalently Algorithm 2) and SDCA. The traditional analysis in SDCA [39][23][40][38] is premised on the assumption that $R(\cdot)$ is a strongly convex function, whereby the first term in the dual objective (10) is a smooth function. In contrast, for our GSFW method $R(\cdot)$ need not have any such structure; indeed in the Frank-Wolfe setting $R(\cdot)$ can be an indicator function of the (primal) feasible region whereby the first term in the dual problem (10) is then non-differentiable. Viewing GSFW through the dual (Algorithm 2), we compute a subgradient of the first part of the dual objective by calling a linear optimization oracle in the primal space, and this subgradient is used in the dual mirror descent algorithm for solving (10). From the SDCA perspective, this is the first such version of SDCA that does not require a strongly convex regularizer $R(\cdot)$.

(As a thought exercise, it is surely possible to start with a non-strongly convex $R(\cdot)$ and then add a tiny strongly convex regularizer term based on a target optimality tolerance and other continuity parameters [though such tuning can be tricky to do in practice], and then use SDCA rather than GSFW. However, in this approach the subproblem that needs to be solved at each iteration requires a projection step onto the feasible region as opposed to solving a linear optimization subproblem, which goes beyond and can be much more computationally demanding than Frank-Wolfe in certain settings. For example, in matrix completion (Example 1.2), solving the linear optimization oracle with the nuclear norm ball requires the computation of the largest eigenvector/eigenvalue pair, while doing the projection requires a full eigendecomposition, and therefore can be significantly more computationally burdensome. Of course, this is one of the key reasons why Frank-Wolfe methods have been so extensively studied in the past decade.)

GSFW has closer connections to SDCA in the case when $R(\cdot)$ is strongly convex – whereby the first term in the dual objective is smooth. In this case Randomized Coordinate Mirror Descent (Algorithm 2) can be viewed as a variant of SDCA with a specific updating rule based on the mirror descent methodology. This perspective provides a new interpretation for SDCA in the primal space as a variant of a Frank-Wolfe based method. In the previous literature involving SDCA, even though one can rewrite SDCA entirely in the primal space [37], there are still explicit dual variables which lack intuition or interpretation in the primal space, in contrast to our Algorithm 1 and Algorithm 2 equivalency.

In Algorithm 2 the coordinate update at each iteration requires the solution of a univariate problem of the following form for a suitably given scalar c_{j_i}:

$$\min_{w_{j_i}} c_{j_i} w_{j_i} + l^*_{j_i}(w_{j_i}) \ , \tag{26}$$

in comparison with the basic version SDCA algorithms in [39] or [23] for which the coordinate update at each iteration requires the solution of the following slightly different univariate problem for suitably given scalars c_{j_i} and b_{j_i}:

$$\min_{w_{j_i}} c_{j_i} w_{j_i} + l^*_{j_i}(w_{j_i}) + b_{j_i} w^2_{j_i} \ . \tag{27}$$

On the other hand, we point out that there are variants of SDCA that do not require solving (27), see the update rules [III], [IV] and [V] in [38], but these rules are still different from (26). Indeed, the unaccelerated version of [23] can be viewed as a randomized coordinate method with a composite function, and has the following update:

$$w^{i+1}_{j_i} = \arg\min_{w_{j_i}} \left\{ c_{j_i} w_{j_i} + \tfrac{1}{2\eta}(w_{j_i} - w^i_{j_i})^2 + \tfrac{1}{n} l^*_{j_i}(w_{j_i}) \right\} \ ,$$

where c_{j_i} is one coordinate of the gradient. The above update can be viewed as a coordinate mirror descent method update with reference function $h(w) := \tfrac{1}{2\eta}\|w\|^2 + \tfrac{1}{n}L^*(w)$. This is similar to the equivalence of composite optimization and mirror descent in the deterministic case discussed in Section 3.3 of [25].

A.5 Regarding Randomized Coordinate Mirror Descent with Non-smooth Functions

Since the seminal work of Nesterov [32], there have been many research results on randomized coordinate descent for convex minimization of a general smooth objective function; however, there has not been much research on randomized coordinate descent in the general non-smooth setting. [31] develops a randomized block-coordinate method for some specially structured problems. Also, many papers consider a composite objective function $f(\cdot) := \hat{f}(\cdot) + \tau(\cdot)$ where $\hat{f}(\cdot)$ is smooth and $\tau(\cdot)$ is non-smooth, separable, and computationally friendly, see in particular [32], [35], [26] among many others. These methods cannot be applied to the general non-smooth case so far as we can tell. We can consider the case where $f(\cdot)$ is the sum of a non-smooth Lipschitz continuous function $\bar{f}(\cdot)$ and a separable strongly convex function, namely $f(w) := \bar{f}(w) + \sum_{j=1}^n \tau_j(w_j)$ where $\tau_j(\cdot) : \mathbb{R} \to \mathbb{R}$ is strongly convex on its domain. Then $f(\cdot)$ is in the format of the dual problem (10), whereby Theorem 3.1 can be applied to show that the randomized coordinate mirror descent method (Algorithm 2) yields the indicated computational guarantees in this case. This shows that non-smooth convex optimization can be tackled by randomized coordinate descent if the objective function is the sum of a non-smooth function and a strongly convex separable function.

References

[1] Zeyuan Allen-Zhu, *Katyusha: The first direct acceleration of stochastic gradient methods*, Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, ACM, 2017, pp. 1200–1205.

[2] Zeyuan Allen-Zhu and Yang Yuan, *Improved SVRG for non-strongly-convex or sum-of-non-convex objectives*, International conference on machine learning, 2016, pp. 1080–1089.

[3] Mordecai Avriel, *Nonlinear optimization: Analysis and methods*, Prentice-Hall, 1976.

[4] Francis Bach, *Duality between subgradient and conditional gradient methods*, SIAM Journal on Optimization **25** (2015), no. 1, 115–129.

[5] Heinz Bauschke, Jérôme Bolte, and Marc Teboulle, *A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications*, Mathematics of Operations Research **42** (2016), no. 2, 330–348.

[6] A. Beck and M. Teboulle, *Mirror descent and nonlinear projected subgradient methods for convex optimization*, Operations Research Letters **31** (2003), no. 3, 167–175.

[7] Emmanuel J Candès and Benjamin Recht, *Exact matrix completion via convex optimization*, Foundations of Computational mathematics **9** (2009), no. 6, 717.

[8] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, ACM transactions on intelligent systems and technology (TIST) **2** (2011), no. 3, 27.

[9] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien, *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, Advances in neural information processing systems, 2014, pp. 1646–1654.

[10] Maryam Fazel, *Matrix rank minimization with applications*, Ph.D. thesis, PhD thesis, Stanford University, 2002.

[11] Marguerite Frank and Philip Wolfe, *An algorithm for quadratic programming*, Naval Research Logistics Quarterly **3** (1956), 95–110.

[12] Robert Freund and Paul Grigas, *New analysis and results for the Frank–Wolfe method*, Mathematical Programming **155** (2016), no. 1-2, 199–230.

[13] Robert Freund, Paul Grigas, and Rahul Mazumder, *An extended Frank–Wolfe method with in-face directions, and its application to low-rank matrix completion*, SIAM Journal on Optimization **27** (2017), no. 1, 319–346.

[14] Filip Hanzely and Peter Richtárik, *Fastest rates for stochastic mirror descent methods*, arXiv preprint arXiv:1803.07374 (2018).

[15] Zaid Harchaoui, Anatoli Juditsky, and Arkadi Nemirovski, *Conditional gradient algorithms for norm-regularized smooth convex optimization*, Mathematical Programming **152** (2015), no. 1-2, 75–112.

[16] Elad Hazan and Satyen Kale, *Projection-free online learning*, arXiv preprint arXiv:1206.4657 (2012).

[17] Elad Hazan and Haipeng Luo, *Variance-reduced and projection-free stochastic optimization*, International Conference on Machine Learning, 2016, pp. 1263–1271.

[18] Arthur E Hoerl and Robert W Kennard, *Ridge regression: Biased estimation for nonorthogonal problems*, Technometrics **12** (1970), no. 1, 55–67.

[19] Martin Jaggi, *Revisiting Frank-Wolfe: Projection-free sparse convex optimization*, Proceedings of the 30th International Conference on Machine Learning (ICML-13), 2013, pp. 427–435.

[20] Rie Johnson and Tong Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, Advances in neural information processing systems, 2013, pp. 315–323.

[21] Sham M. Kakade, Shai Shalev-Shwartz, and Ambuj Tewari, *Regularization techniques for learning with matrices*, The Journal of Machine Learning Research **13** (2012), no. 1, 1865–1890.

[22] Guanghui Lan and Yi Zhou, *Conditional gradient sliding for convex optimization*, SIAM Journal on Optimization **26** (2016), no. 2, 1379–1409.

[23] Qihang Lin, Zhaosong Lu, and Lin Xiao, *An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization*, SIAM Journal on Optimization **25** (2015), no. 4, 2244–2273.

[24] Haihao Lu, *"Relative-continuity" for non-lipschitz non-smooth convex optimization using stochastic (or deterministic) mirror descent*, arXiv preprint arXiv:1710.04718 (2017).

[25] Haihao Lu, Robert Freund, and Yurii Nesterov, *Relatively smooth convex optimization by first-order methods, and applications*, SIAM Journal on Optimization **28** (2018), no. 1, 333–354.

[26] Zhaosong Lu and Lin Xiao, *On the complexity analysis of randomized block-coordinate descent methods*, Mathematical Programming **152** (2015), no. 1-2, 615–642.

[27] Michael Mahoney and Petros Drineas, *CUR matrix decompositions for improved data analysis*, Proceedings of the National Academy of Sciences **106** (2009), no. 3, 697–702.

[28] Julien Mairal, *Incremental majorization-minimization optimization with application to large-scale machine learning*, SIAM Journal on Optimization **25** (2015), no. 2, 829–855.

[29] Julien Mairal, Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach, *Convex and network flow optimization for structured sparsity*, Journal of Machine Learning Research **12** (2011), no. Sep, 2681–2720.

[30] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi, *Stochastic conditional gradient methods: From convex minimization to submodular maximization*, arXiv preprint arXiv:1804.09554 (2018).

[31] Yu Nesterov, *Subgradient methods for huge-scale optimization problems*, Mathematical Programming **146** (2014), no. 1-2, 275–297.

[32] Yurii Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM Journal on Optimization **22** (2012), no. 2, 341–362.

[33] Zheng Qu, Peter Richtárik, and Tong Zhang, *Quartz: Randomized dual coordinate ascent with arbitrary sampling*, Advances in neural information processing systems, 2015, pp. 865–873.

[34] Pradeep Ravikumar, Martin Wainwright, and John Lafferty, *High-dimensional ising model selection using 1-regularized logistic regression*, The Annals of Statistics **38** (2010), no. 3, 1287–1319.

[35] Peter Richtarik and Martin Takac, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Mathematical Programming **144** (2014), no. 1-2, 1–38.

[36] Mark Schmidt, Nicolas Le Roux, and Francis Bach, *Minimizing finite sums with the stochastic average gradient*, Mathematical Programming **162** (2017), no. 1-2, 83–112.

[37] Shai Shalev-Shwartz, *Sdca without duality, regularization, and individual convexity*, International Conference on Machine Learning, 2016, pp. 747–754.

[38] Shai Shalev-Shwartz and Tong Zhang, *Proximal stochastic dual coordinate ascent*, arXiv preprint arXiv:1211.2717 (2012).

[39] _____, *Stochastic dual coordinate ascent methods for regularized loss minimization*, Journal of Machine Learning Research **14** (2013), no. Feb, 567–599.

[40] _____, *Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization*, International Conference on Machine Learning, 2014, pp. 64–72.

[41] Robert Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society. Series B (Methodological) (1996), 267–288.

[42] Paul Tseng, *On accelerated proximal gradient methods for convex-concave optimization*, Tech. report, May 21, 2008.

[43] _____, *On accelerated proximal gradient methods for convex-concave optimization*, Tech. report, May 21, 2008.

[44] Quang Van Nguyen, *Forward-backward splitting with Bregman distances*, Vietnam Journal of Mathematics **45** (2017), no. 3, 519–539.

[45] Yaoliang Yu, Xinhua Zhang, and Dale Schuurmans, *Generalized conditional gradient for sparse estimation*, The Journal of Machine Learning Research **18** (2017), no. 1, 5279–5324.

[46] Constantin Zalinescu, *Convex analysis in general vector spaces*, World Scientific, 2002.

[47] Yuchen Zhang and Lin Xiao, *Stochastic primal-dual coordinate method for regularized empirical risk minimization*, The Journal of Machine Learning Research **18** (2017), no. 1, 2939–2980.