JMLR: Workshop and Conference Proceedings vol 49:1–43, 2016

A Light Touch for Heavily Constrained SGD

Andrew Cotter ACOTTER@GOOGLE.COM
Maya Gupta MAYAGUPTA@GOOGLE.COM
Jan Pfeifer JANPF@GOOGLE.COM
1600 Amphitheatre Parkway
Mountain View, CA 94043

Abstract

Minimizing empirical risk subject to a set of constraints can be a useful strategy for learning restricted classes of functions, such as monotonic functions, submodular functions, classifiers that guarantee a certain class label for some subset of examples, etc. However, these restrictions may result in a very large number of constraints. Projected stochastic gradient descent (SGD) is often the default choice for large-scale optimization in machine learning, but requires a projection after each update. For heavily-constrained objectives, we propose an efficient extension of SGD that stays close to the feasible region while only applying constraints probabilistically at each iteration. Theoretical analysis shows a compelling trade-off between per-iteration work and the number of iterations needed on problems with a large number of constraints.

1. Introduction

Many machine learning problems can benefit from the addition of constraints. For example, one can learn monotonic functions by adding appropriate constraints to ensure or encourage positive derivatives everywhere (e.g. Archer and Wang, 1993; Sill, 1998; Spouge et al., 2003; Daniels and Velikova, 2010; Gupta et al., 2016). Submodular functions can often be learned from noisy examples by imposing constraints to ensure submodularity holds. Another example occurs when one wishes to guarantee that a classifier will correctly label certain “canonical” examples, which can be enforced by constraining the function values on those examples. See Qu and Hu (2011) for some other examples of constraints useful in machine learning.

However, these practical uses of constraints in machine learning are impractical in that the number of constraints may be very large, and scale poorly with the number of features d or number of training samples n. In this paper we propose a new strategy for tackling such heavily-constrained problems, with guarantees and compelling convergence rates for large-scale convex problems.

A standard approach for large-scale empirical risk minimization is projected stochastic gradient descent (e.g. Zinkevich, 2003; Nemirovski et al., 2009). Each SGD iteration is computationally cheap, and the algorithm converges quickly to a solution good enough for machine learning needs. However, this algorithm requires a projection onto the feasible region after each stochastic gradient step, which can be prohibitively slow if there are many non-trivial constraints, and is not easy to parallelize. Recently, Frank-Wolfe-style algorithms (e.g. Hazan and Kale, 2012; Jaggi, 2013) have been proposed that remove the projection, but require a constrained linear optimization at each iteration.

We propose a new strategy for large-scale constrained optimization that, like Mahdavi et al. (2012), moves the constraints into the objective and finds an approximate solution of the resulting unconstrained problem, projecting the (potentially-infeasible) result onto the constraints only once, at the end. Their work focused on handling only one constraint, but as they noted, multiple constraints $g_1(x) \leq 0, g_2(x) \leq 0, \ldots, g_m(x) \leq 0$ can be reduced to one constraint by replacing the m constraints with their maximum: $\max_i g_i(x) \leq 0$. However, this still requires that all m constraints be checked at every iteration. In this paper, we focus on the computational complexity as a function of the number of constraints m, and show that it is possible to achieve good convergence rates without checking constraints so often.

The key challenge to handling a large number of constraints is determining which constraints are active at the optimum of the constrained problem, which is likely to be only a small fraction of the total constraint set. For example, for linear inequality constraints on a d-dimensional problem, no more than d of the constraints will be active at the optimum, and furthermore, once the active constraints are known, the problem reduces to solving the unconstrained problem that results from projecting onto them, which is typically vastly easier.

To identify and focus on the important constraints, we propose learning a probability distribution over the m constraints that concentrates on the most-violated, and *sampling* constraints from this evolving distribution at each iteration. We call this approach LightTouch because at each iteration only a few constraints are checked, and the solution is only nudged toward the feasible set. LightTouch is suitable for convex problems, but we also propose a variant, MidTouch, that enjoys a superior convergence rate on strongly convex problems. These two algorithms are introduced and analyzed in Section 3.

Our proposed strategy removes the per-iteration m-dependence on the number of constraint evaluations. LightTouch and MidTouch do need more iterations to converge, but each iteration is faster, resulting in a net performance improvement. To be precise, we show that the total number of constraint checks required to achieve ϵ-suboptimality when optimizing a non-strongly convex objective decreases from $O(m/\epsilon^2)$ to $\tilde{O}((\ln m)/\epsilon^2 + m(\ln m)^{3/2}/\epsilon^{3/2})$—notice that the m-dependence of the dominant (in ϵ) term has decreased from m to $\ln m$. For a λ-strongly convex objective, the dominant (again in ϵ) term in our bound on the number of constraint checks decreases from $O(m/\lambda^2\epsilon)$ to $\tilde{O}((\ln m)/\lambda^2\epsilon)$, but like the non-strongly convex result this bound contains lower-order terms with worse m-dependencies. A more careful comparison of the performance of our algorithms can be found in Section 4.

While they check fewer than m constraints per iteration, these algorithms do need to pay a $O(m)$ per-iteration *arithmetic* cost. When each constraint is expensive to check, this cost can be neglected. However, when the constraints are simple to check (e.g. box constraints, or the lattice monotonicity constraints considered in our experiments), it can be partially addressed by transforming the problem into an equivalent one with fewer more costly constraints. This, as well as other practical considerations, are discussed in Section 5.

Experiments on a large-scale real-world heavily-constrained ranking problem show that our proposed approach works well in practice. This problem was too large for a projected SGD implemen-

Table 1: Key notation.

Symbol	Description	Definition
$\mathcal{W}$	Bounded, closed and convex domain	$\mathcal{W} \subseteq \mathbb{R}^d$
Δ^m	m-dimensional simplex	$\Delta^m = \{p \in \mathbb{R}^m \mid p_i \geq 0 \wedge \sum_{i=1}^m p_i = 1\}$
d	Dimension of $\mathcal{W}$	
m	Number of constraints	
f	Unconstrained objective function	$f : \mathcal{W} \rightarrow \mathbb{R}$
g_i	Convex constraint functions	$g_i : \mathcal{W} \rightarrow \mathbb{R}$
g	Combined constraint function	$g(w) = \max_i g_i(w)$
Π_w	Projection onto $\mathcal{W}$	$\Pi_w(w) = \mathrm{argmin}_{\{w' \in \mathcal{W}\}} \lVert w - w' \rVert_2$
Π_p	Projection onto Δ^m	$\Pi_p(p) = p / \lVert p \rVert_1$
Π_g	Projection onto constraints	$\Pi_g(w) = \mathrm{argmin}_{\{w' \in \mathcal{W} : g(w') \leq 0\}} \lVert w - w' \rVert_2$
ρ	Boundary gradient magnitude	If $g(w) = 0$, then $\rho \leq \lVert \check{\nabla} \rVert_2$ for all $\check{\nabla} \in \partial g(w)$
γ	Constraint scaling factor	$\gamma > L_f / \rho$
h	Objective function	$h(w) = f(w) + \gamma \max(0, g(w))$
$\tilde{h}$	Relaxed objective function	$\tilde{h}(w, p) = f(w) + \gamma \sum_{i=1}^m p_i \max(0, g_i(w))$
L_f	Lipschitz constant of f	$L_f \lVert w - w' \rVert_2 \geq \lvert f(w) - f(w') \rvert$
L_g	Lipschitz constant of the g_is	$L_g \lVert w - w' \rVert_2 \geq \lvert g_i(w) - g_i(w') \rvert$
D_w	Bound (≥ 1) on diameter of $\mathcal{W}$	$D_w \geq \sup_{w, w' \in \mathcal{W}} \max\{1, \lVert w - w' \rVert_2\}$
G_f	Bound on stochastic subgradients of f	$G_f \geq \lVert \check{\Delta}^{(t)} \rVert_2$
G_g	Bound on stochastic subgradients of g_is	$G_g \geq \lVert \check{\nabla} \max(0, g_i(w)) \rVert_2$
$\check{\Delta}$	Stochastic subgradient of f	
$\check{\Delta}_w$	Stochastic subgradient of $\tilde{h}$ w.r.t. w	
$\hat{\Delta}_p$	Stochastic supergradient of $\tilde{h}$ w.r.t. p	
μ	Remembered gradient coordinates (Johnson and Zhang, 2013)	
k	Minibatch size in LightTouch's p-update	
$\bar{w}$	Average iterate	$\bar{w} = (\sum_{t=1}^T w^{(t)}) / T$

tation using an off-the-shelf quadratic programming solver to perform projections, but was amenable to an approach based on a fast approximate projection routine tailored to this particular constraint set. Measured in terms of runtime, however, LightTouch was still significantly faster. Each constraint in this problem is trivial, requiring only a single comparison operation to check, so the aforementioned $O(m)$ arithmetic cost of LightTouch is a significant issue. Despite this, LightTouch was roughly as fast as the Mahdavi et al. (2012)-like algorithm FullTouch. In light of other experiments showing that LightTouch checks dramatically fewer constraints in total than FullTouch, we believe that LightTouch is well-suited to machine learning problems with many nontrivial constraints.

2. Heavily Constrained SGD

Consider the constrained optimization problem:

$$\begin{aligned} &\min_{w \in \mathcal{W}} f(w) \\ &\text{s.t. } g_i(w) \leq 0 \;\; \forall i \in \{1, \ldots, m\}, \end{aligned} \tag{1}$$

where $\mathcal{W} \subseteq \mathbb{R}^d$ is bounded, closed and convex, and $f : \mathcal{W} \rightarrow \mathbb{R}$ and all $g_i : \mathcal{W} \rightarrow \mathbb{R}$ are convex (our notation is summarized in Table 1). We assume that $\mathcal{W}$ is a simple object, e.g. an ℓ^2 ball, onto

Algorithm 1 (FullTouch) Minimizes f on $\mathcal{W}$ subject to the single constraint $g(w) \leq 0$. For problems with m constraints $g_i(w) \leq 0$, let $g(w) = \max_i g_i(w)$, in which case differentiating $\max\{0, g(w)\}$ (line 4) requires evaluating all m constraints. This algorithm—our starting point—is similar to those proposed by Mahdavi et al. (2012), and like their algorithms only contains a single projection, at the end, projecting the potentially-infeasible result vector $\bar{w}$.

```
  Hyperparameters: $T$, $\eta$
1   Initialize $w^{(1)} \in \mathcal{W}$ arbitrarily
2   For $t = 1$ to $T$:
3       Sample $\check{\Delta}^{(t)}$        // stochastic subgradient of $f(w^{(t)})$
4       Let $\check{\Delta}_w^{(t)} = \check{\Delta}^{(t)} + \gamma \check{\nabla} \max\{0, g(w^{(t)})\}$
5       Update $w^{(t+1)} = \Pi_w(w^{(t)} - \eta \check{\Delta}_w^{(t)})$        // $\Pi_w$ projects its argument onto $\mathcal{W}$ w.r.t. $\|\cdot\|_2$
6   Average $\bar{w} = (\sum_{t=1}^{T} w^{(t)})/T$
7   Return $\Pi_g(\bar{w})$        // optional if small constraint violations are acceptable
```

which it is inexpensive to project, and that the "trickier" aspects of the domain are specified via the constraints $g_i(w) \leq 0$. Notice that we consider constraints written in terms of arbitrary convex functions, and are not restricted to e.g. only linear or quadratic constraints.

2.1. FullTouch: A Relaxation with a Feasible Minimizer

We build on the approach of Mahdavi et al. (2012) to relax Equation 1. Defining $g(w) = \max_i g_i(w)$ and introducing a Lagrange multiplier α yields the equivalent optimization problem:

$$\max_{\alpha \geq 0} \min_{w \in \mathcal{W}} f(w) + \alpha g(w) . \tag{2}$$

Directly optimizing over w and α is problematic because the optimal value for α is infinite for any w that violates a constraint. Instead, we follow Mahdavi et al. (2012, Section 4.2) in relaxing the problem by adding an upper bound of γ on α, and using the fact that $\max_{0 \leq \alpha \leq \gamma} \alpha g(w) = \gamma \max(0, g(w))$.

In the following lemma, we show that, with the proper choice of γ, any minimizer of this relaxed objective is a feasible solution of Equation 1, indicating that using stochastic gradient descent (SGD) to minimize the relaxation ($h(w)$ in the lemma below) will be effective.

Lemma 1 *Suppose that f is L_f-Lipschitz, i.e. $|f(w) - f(w')| \leq L_f \|w - w'\|_2$ for all $w, w' \in \mathcal{W}$, and that there is a constant $\rho > 0$ such that if $g(w) = 0$ then $\|\check{\nabla}\|_2 \geq \rho$ for all $\check{\nabla} \in \partial g(w)$, where $\partial g(w)$ is the subdifferential of $g(w)$.*

For a parameter $\gamma > 0$, define:

$$h(w) = f(w) + \gamma \max\{0, g(w)\} .$$

If $\gamma > L_f / \rho$, then for any infeasible w (i.e. for which $g(w) > 0$):

$$h(w) > h(\Pi_g(w)) = f(\Pi_g(w)) \qquad \text{and} \qquad \|w - \Pi_g(w)\|_2 \leq \frac{h(w) - h(\Pi_g(w))}{\gamma\rho - L_f},$$

where $\Pi_g(w)$ is the projection of w onto the set $\{w \in \mathcal{W} : g(w) \leq 0\}$ w.r.t. the Euclidean norm.

Proof In Appendix C. ■

The strategy of applying SGD to $h(w)$, detailed in Algorithm 1, which we call FullTouch, has the same "flavor" as the algorithms proposed by Mahdavi et al. (2012), and we use it as a baseline comparison point for our other algorithms.

Application of a standard SGD bound to FullTouch shows that it converges at a rate with no explicit dependence on the number of constraints m, measured in terms of the number of iterations required to achieve some desired suboptimality (see Appendix C.1), although the γ parameter can introduce an *implicit* d or m-dependence, depending on the constraints (discussed in Section 2.2). The main drawback of FullTouch is that each iteration is expensive, requiring the evaluation of all m constraints, since differentiation of g requires first identifying the most-violated. This is the key issue we tackle with the LightTouch algorithm proposed in Section 3.

2.2. Constraint-Dependence of γ

The conditions on Lemma 1 were stated in terms of g, instead of the individual g_is, because it is difficult to provide suitable conditions on the "component" constraints without accounting for their interactions.

For a point w where two or more constraints intersect, the subdifferential of $g(w)$ consists of all convex combinations of subgradients of the intersecting constraints, with the consequence that even if each of the subgradients of the $g_i(w)$s has norm at least ρ', subgradients of $g(w)$ will generally have norms smaller than ρ'. Exactly how much smaller depends on the particular constraints under consideration. We illustrate this phenomenon with the following examples, but note that, in practice, γ should be chosen experimentally for any particular problem, so the question of the d and m-dependence of γ is mostly of theoretical interest.

Box Constraints Consider the $m = 2d$ box constraints $g_i(w) = -w_i - 1$ and $g_{i+d}(w) = w_i - 1$, all of which have gradients of norm 1. At most d constraints can intersect (at a corner of the $[-1, 1]^d$ box), all of which are mutually orthogonal, so the norm of any convex combination of their gradients is lower bounded by that of their average, $\rho = 1/\sqrt{d}$. Hence, one should choose $\gamma > \sqrt{d}\, L_f$.

As in the above example, $\gamma \propto \sqrt{\min(m, d)}$ will suffice when the subgradients of intersecting constraints are at least orthogonal, and γ can be smaller if they always have positive inner products. However, if subgradients of intersecting constraints tend to point in opposing directions, then γ may need to be much larger, as in our next example:

Ordering Constraints Suppose the $m = d - 1$ constraints order the components of w as $w_1 \leq w_2 \leq \cdots \leq w_d$, for which $g_i(w) = (w_i - w_{i+1})/\sqrt{2}$, gradients of which again have norm 1. All of these constraints may be active simultaneously, in which case there is widespread cancellation in the average gradient $(e_1 - e_d)/(m\sqrt{2})$, where e_i is the ith standard unit basis vector. The norm of this average gradient is $\rho = 1/m$, so we should choose $\gamma > (d - 1)L_f$.

In light of this example, one begins to wonder if a suitable γ will necessarily *exist*—fortunately, the convexity of g enables us to prove a trivial bound as long as $g(v)$ is strictly negative for some $v \in \mathcal{W}$:

Lemma 2 *Suppose that there exists a* $v \in \mathcal{W}$ *for which* $g(v) < 0$*, and let* $D_w \geq \sup_{w,w' \in \mathcal{W}} \left\| w - w' \right\|_2$ *bound the diameter of* $\mathcal{W}$*. Then* $\rho = -g(v)/D_w$ *satisfies the conditions of Lemma 1.*

Proof Let $w \in \mathcal{W}$ be a point for which $g(w) = 0$, and $\check{\nabla} \in \partial g(w)$ an arbitrary subgradient. By convexity, $g(v) \geq g(w) + \langle v - w, \check{\nabla} \rangle$. The Cauchy-Schwarz inequality then gives that:

$$g(v) \geq - \left\| v - w \right\|_2 \left\| \check{\nabla} \right\|_2 ,$$

from which the claim follows immediately. ■

Linear Constraints Consider the constraints $Aw \preceq b$, with each row of A having unit norm, $b_{\min} = \min_i b_i > 0$, and $\mathcal{W}$ being the ℓ^2 ball of radius r. It follows from Lemma 2 that $\gamma > (2r/b_{\min})L_f$ suffices. Notice that the earlier box constraint example satisfies these assumptions (with $b_{\min} = 1$ and $r = \sqrt{d}$).

As the above examples illustrate, subgradients of g will be large at the boundary if subgradients of the g_is are large, *and* the constraints intersect at sufficiently shallow angles that, representing boundary subgradients of g as convex combinations of subgradients of the g_is, the components reinforce each other, or at least do not cancel *too* much. This requirement is related to the linear regularity assumption introduced by Bauschke (1996), and considered recently by Wang et al. (2015).

3. A Light Touch

This section presents the main contribution of this paper: an algorithm that stochastically samples a small subset of the m constraints at each SGD iteration, updates the parameters based on the subgradients of the sampled constraints, and carefully learns the distribution over the constraints to produce a net performance gain.

We first motivate the approach by considering an oracle, then explain the algorithm and present convergence results for the convex (Section 3.2) and strongly convex (Section 3.3) cases.

3.1. Wanted: An Oracle For the Most Violated Constraint

Because FullTouch only needs to differentiate the most violated constraint at each iteration, it follows that if one had access to an oracle that identified the most-violated constraint, then the overall convergence rate (including the cost of each iteration) could *only* depend on m through γ. This motivates us to *learn* to predict the most-violated constraint, ideally at a significantly better than linear-in-m rate.

Algorithm 2 (LightTouch) Minimizes f on $\mathcal{W}$ subject to the constraints $g_i(w) \leq 0$ for $i \in \{1, \dots, m\}$. The algorithm learns an auxiliary probability distribution p (lines 9–13) estimating how likely it is that each constraint is the most-violated. We assume that $k \leq m$: if $k > m$, then the user is willing to check m constraints per iteration *anyway*, so FullTouch is the better choice. Like FullTouch, this algorithm finds a potentially-infeasible solution $\bar{w}$ which is only projected onto the feasible region at the end. Notice that while the p-update checks only k constraints, it does require $O(m)$ arithmetic operations. This issue is discussed further in Section 5.1.

```
    Hyperparameters: $T$, $\eta$, $k$
1       Initialize $w^{(1)} \in \mathcal{W}$ arbitrarily
2       Initialize $p^{(1)} \in \Delta^m$ to the uniform distribution
3       Initialize $\mu_j^{(1)} = \max\{0, g_j(w^{(1)})\}$        // 0 if $w^{(1)}$ is feasible
4       For $t = 1$ to $T$:
5           Sample $\check{\Delta}^{(t)}$        // stochastic subgradient of $f(w^{(t)})$
6           Sample $i^{(t)} \sim p^{(t)}$
7           Let $\check{\Delta}_w^{(t)} = \check{\Delta}^{(t)} + \gamma \check{\nabla} \max\{0, g_{i^{(t)}}(w^{(t)})\}$
8           Update $w^{(t+1)} = \Pi_w(w^{(t)} - \eta \check{\Delta}_w^{(t)})$        // $\Pi_w$ projects its argument onto $\mathcal{W}$ w.r.t. $\|\cdot\|_2$
9           Sample $S^{(t)} \subseteq \{1, \dots, m\}$ with $|S^{(t)}| = k$ uniformly without replacement
10          Let $\hat{\Delta}_p^{(t)} = \gamma \mu^{(t)} + (\gamma m / k) \sum_{j \in S^{(t)}} e_j(\max\{0, g_j(w^{(t)})\} - \mu_j^{(t)})$
11          Let $\mu_j^{(t+1)} = \max\{0, g_j(w^{(t)})\}$ if $j \in S^{(t)}$, otherwise $\mu_j^{(t+1)} = \mu_j^{(t)}$
12          Update $\tilde{p}^{(t+1)} = \exp(\ln p^{(t)} + \eta \hat{\Delta}_p^{(t)})$        // element-wise $\exp$ and $\ln$
13          Project $p^{(t+1)} = \tilde{p}^{(t+1)} / \|\tilde{p}^{(t+1)}\|_1$
14      Average $\bar{w} = (\sum_{t=1}^{T} w^{(t)}) / T$
15      Return $\Pi_g(\bar{w})$        // optional if small constraint violations are acceptable
```

To this end, we further relax the problem of minimizing $h(w)$ (defined in Lemma 1) by replacing $\gamma \max(0, g(w))$ with maximization over a probability distribution (as in Clarkson et al. (2010)), yielding the equivalent convex-linear optimization problem:

$$\max_{p \in \Delta^m} \min_{w \in \mathcal{W}} \tilde{h}\left(w, p\right) \tag{3}$$

$$\text{where } \tilde{h}\left(w, p\right) = f\left(w\right) + \gamma \sum_{i=1}^{m} p_i \max\left\{0, g_i\left(w\right)\right\}.$$

Here, Δ^m is the m-dimensional simplex. We propose optimizing over w and p jointly, thereby learning the most-violated constraint, represented by the multinoulli distribution p over constraint indices, at the same time as we optimize over w.

3.2. LightTouch: Stochastic Constraint Handling

To optimize Equation 3, our proposed algorithm (Algorithm 2, LightTouch) iteratively samples stochastic gradients $\check{\Delta}_w^{(t)}$ w.r.t. w and $\hat{\Delta}_p^{(t)}$ w.r.t. p of $\tilde{h}(w, p)$, and then takes an SGD step on w and a multiplicative step on p:

$$w^{(t+1)} = \Pi_w\left(w^{(t)} - \eta \check{\Delta}_w^{(t)}\right) \qquad \text{and} \qquad p^{(t+1)} = \Pi_p\left(\exp\left(\ln p^{(t)} + \eta \hat{\Delta}_p^{(t)}\right)\right),$$

where the $\exp$ and $\ln$ of the p-update are performed element-wise, Π_w projects onto $\mathcal{W}$ w.r.t. the Euclidean norm, and Π_p onto Δ^m via normalization (i.e. dividing its parameter by its sum).

The key to getting a good convergence rate for this algorithm is to choose $\check{\Delta}_w$ and $\hat{\Delta}_p$ such that they are both inexpensive to compute, and tend to have small norms. For $\check{\Delta}_w$, this can be accomplished straightforwardly, by sampling a constraint index i according to p, and taking:

$$\check{\Delta}_w = \check{\Delta} + \gamma \check{\nabla} \max \left\{0, g_i\left(w\right)\right\},$$

where $\check{\Delta}$ is a stochastic subgradient of f and $\check{\nabla} \max(0, g_i(w))$ is a subgradient of $\max(0, g_i(w))$. Calculating each such $\check{\Delta}_w$ requires differentiating only one constraint, and it is easy to verify that $\check{\Delta}_w$ is a subgradient of $\tilde{h}$ w.r.t. w in expectation over $\check{\Delta}$ and i. Taking G_f to be a bound on the norm of $\check{\Delta}$ and G_g on the norms of subgradients of the g_is shows that $\check{\Delta}_w$'s norm is bounded by $G_f + \gamma G_g$.

For $\hat{\Delta}_p$, some care must be taken. Simply sampling a constraint index j uniformly and defining:

$$\hat{\Delta}_p = \gamma m e_j \max \left\{0, g_j\left(w\right)\right\},$$

where e_j is the jth m-dimensional standard unit basis vector, does produce a $\hat{\Delta}_p$ that in expectation is the gradient of $\tilde{h}$ w.r.t. p, but it has a norm bound proportional to m. Such potentially large stochastic gradients would result in the number of iterations required to achieve some target suboptimality being proportional to m^2 in our final bound.

A typical approach to reducing the variance (and hence the expected magnitude) of $\hat{\Delta}_p$ is minibatching: instead of sampling a single constraint index j at every iteration, we could instead sample a subset S of size $|S| = k$ without replacement, and use:

$$\hat{\Delta}_p = \frac{\gamma m}{k} \sum_{j \in S} e_j \max \left\{0, g_j\left(w\right)\right\}.$$

This is effective, but not enough, because reducing the variance by a factor of k via minibatching requires that we check k times more constraints. For this reason, in addition to minibatching, we center the stochastic gradients, as is done by the well-known SVRG algorithm (Johnson and Zhang, 2013), by storing a gradient estimate $\gamma\mu$ with $\mu \in \mathbb{R}^m$, at each iteration sampling a set S of size $|S| = k$ uniformly without replacement, and computing:

$$\hat{\Delta}_p = \gamma\mu + \frac{\gamma m}{k} \sum_{j \in S} e_j \left(\max\left\{0, g_j(w)\right\} - \mu_j\right). \tag{4}$$

We then update the jth coordinate of μ to be $\mu_j = \max\{0, g_j(w)\}$ for every $j \in S$. The norms of the resulting stochastic gradients will be small if $\gamma\mu$ is a good estimate of the gradient, i.e. $\mu_j \approx \max(0, g_j(w))$.

The difference between μ_j and $\max(0, g_j(w))$ can be bounded in terms of how many consecutive iterations may have elapsed since μ_j was last updated. It turns out (see Lemma 14 in Appendix C.2) that this quantity can be bounded uniformly by $O((m/k)\ln(mT))$ with high probability, which implies that if the g_is are L_g-Lipschitz, then $|g_j(w) - \mu_j| \leq L_g \eta (G_f + \gamma G_g) O((m/k)\ln(mT))$, since at most $O((m/k)\ln(mT))$ updates of magnitude $\eta(G_f + \gamma G_g)$ may have occurred since μ_j

was last updated. Choosing $\eta \propto 1/\sqrt{T}$, as is standard, moves this portion (the "variance portion") of the $\hat{\Delta}_p$-dependence out of the dominant $O(1/\sqrt{T})$ term and into a subordinate term in our final bound.

The remainder of the $\hat{\Delta}_p$-dependence (the "mean portion") depends on the norm of $\mathbb{E}[\hat{\Delta}_p] = \gamma \sum_j e_j \max(0, g_j(w))$. It is here that our use of multiplicative p-updates becomes significant, because with such updates the relevant norm is the ℓ^∞ norm, instead of e.g. the ℓ^2 norm (as would be the case if we updated p using SGD), thus we can bound $\left\|\mathbb{E}[\hat{\Delta}_p]\right\|_\infty$ with no explicit m-dependence.

The following theorem on the convergence rate of LightTouch is proved by applying a mirror descent bound for saddle point problems while bounding the stochastic gradient norms as described above.

Theorem 3 *Suppose that the conditions of Lemma 1 apply, with $g(w) = \max_i(g_i(w))$. Define $D_w \geq \max\{1, \|w - w'\|_2\}$ as a bound on the diameter of $\mathcal{W}$ (notice that we also choose D_w to be at least 1), $G_f \geq \left\|\check{\Delta}^{(t)}\right\|_2$ and $G_g \geq \left\|\check{\nabla} \max(0, g_i(w))\right\|_2$ as uniform upper bounds on the (stochastic) gradient magnitudes of f and the g_is, respectively, for all $i \in \{1, \dots, m\}$ and $w, w' \in \mathcal{W}$. We also assume that all g_is are L_g-Lipschitz w.r.t. $\|\cdot\|_2$, i.e. $|g_i(w) - g_i(w')| \leq L_g \|w - w'\|_2$. Our result will be expressed in terms of a total iteration count T_ϵ satisfying:*

$$T_\epsilon = O\left(\frac{(\ln m)\, D_w^2 \left(G_f + \gamma G_g + \gamma L_g D_w\right)^2 \ln \frac{1}{\delta}}{\epsilon^2}\right).$$

Define:

$$k = \left\lceil \frac{m\,(1 + \ln m)^{3/4} \sqrt{1 + \ln \frac{1}{\delta}} \sqrt{1 + \ln T_\epsilon}}{T_\epsilon^{1/4}} \right\rceil.$$

If $k \leq m$, then we optimize Equation 1 using T_ϵ iterations of Algorithm 2 (LightTouch), basing the stochastic gradients w.r.t. p on k constraints at each iteration, and using the step size:

$$\eta = \frac{\sqrt{1 + \ln m} D_w}{\left(G_f + \gamma G_g + \gamma L_g D_w\right)\sqrt{T_\epsilon}}.$$

If $k > m$, then LightTouch would check more than m constraints per iteration anyway, so we instead use T_ϵ iterations of Algorithm 1 (FullTouch) with the step size:

$$\eta = \frac{D_w}{\left(G_f + \gamma G_g\right)\sqrt{T_\epsilon}}.$$

In either case, we perform T_ϵ iterations, requiring a total of C_ϵ "constraint checks" (evaluations or differentiations of a single g_i):

$$C_\epsilon = \tilde{O}\left(\frac{(\ln m)\, D_w^2 \left(G_f + \gamma G_g + \gamma L_g D_w\right)^2 \ln \frac{1}{\delta}}{\epsilon^2} + \frac{m\,(\ln m)^{3/2} D_w^{3/2} \left(G_f + \gamma G_g + \gamma L_g D_w\right)^{3/2} \left(\ln \frac{1}{\delta}\right)^{5/4}}{\epsilon^{3/2}}\right).$$

and with probability $1 - \delta$*:*

$$f\left(\Pi_g\left(\bar{w}\right)\right) - f\left(w^*\right) \leq h\left(\bar{w}\right) - h\left(w^*\right) \leq \epsilon \qquad \textit{and} \qquad \left\|\bar{w} - \Pi_g\left(\bar{w}\right)\right\|_2 \leq \frac{\epsilon}{\gamma\rho - L_f},$$

where $w^* \in \{w \in \mathcal{W} : \forall i . g_i(w) \leq 0\}$ *is an arbitrary constraint-satisfying reference vector.*

Proof In Appendix C.2. ■

The most important thing to notice about this theorem is that the dominant terms in the bounds on the number of iterations and number of constraint checks are roughly $\gamma^2 \ln m$ times the usual $1/\epsilon^2$ convergence rate for SGD on a non-strongly convex objective. The lower-order terms have a worse m-dependence, however, with the result that, as the desired suboptimality ϵ shrinks, the algorithm performs fewer constraint checks per iteration until ultimately (once ϵ is on the order of $1/m^2$) only a constant number are checked during each iteration.

3.3. MidTouch: Strong Convexity

To this point, we have only required that the objective function f be convex. However, roughly the same approach also works when f is taken to be λ-strongly convex, although we have only succeeded in proving an in-expectation result, and the algorithm, Algorithm 3 (MidTouch), differs from LightTouch not only in that the w updates use a $1/\lambda t$ step size, but also in being a two-phase algorithm, the first of which, like FullTouch, checks every constraint at each iteration, and the second of which, like LightTouch with $k = 1$, checks only two. The following theorem bounds the convergence rate if we perform $T_1 \approx m\tau^2$ iterations in the first phase and $T_2 \approx \tau^3$ in the second, where the parameter τ determines the total number of iterations performed:

Theorem 4 *Suppose that the conditions of Lemma 1 apply, with* $g(w) = \max_i(g_i(w))$*. Define* $G_f \geq \left\|\check{\Delta}^{(t)}\right\|_2$ *and* $G_g \geq \left\|\check{\nabla} \max(0, g_i(w))\right\|_2$ *as uniform upper bounds on the (stochastic) gradient magnitudes of* f *and the* g_i*s, respectively, for all* $i \in \{1, \dots, m\}$*. We also assume that* f *is* λ*-strongly convex, and that all* g_i*s are* L_g*-Lipschitz w.r.t.* $\|\cdot\|_2$*, i.e.* $|g_i(w) - g_i(w')| \leq L_g \|w - w'\|_2$ *for all* $w, w' \in \mathcal{W}$*. If we run Algorithm 3 (MidTouch) with the* p*-update step size* $\eta = \lambda / 2\gamma^2 L_g^2$ *for* $T_{\epsilon 1}$ *iterations in the first phase and* $T_{\epsilon 2}$ *in the second:*

$$\begin{aligned}
T_{\epsilon 1} =& \tilde{O}\left(\frac{m\left(\ln m\right)^{2/3}\left(G_f + \gamma G_g + \gamma L_g\right)^{4/3}}{\lambda^{4/3}\epsilon^{2/3}} + \frac{m^2\left(\ln m\right)\left(G_f + \gamma G_g\right)}{\lambda\sqrt{\epsilon}}\right), \\
T_{\epsilon 2} =& \tilde{O}\left(\frac{\left(\ln m\right)\left(G_f + \gamma G_g + \gamma L_g\right)^2}{\lambda^2\epsilon} + \frac{m^{3/2}\left(\ln m\right)^{3/2}\left(G_f + \gamma G_g\right)^{3/2}}{\lambda^{3/2}\epsilon^{3/4}}\right),
\end{aligned}$$

requiring a total of C_ϵ *"constraint checks" (evaluations or differentiations of a single* g_i*):*

$$\begin{aligned}
C_\epsilon =& \tilde{O}\left(\frac{\left(\ln m\right)\left(G_f + \gamma G_g + \gamma L_g\right)^2}{\lambda^2\epsilon} + \frac{m^{3/2}\left(\ln m\right)^{3/2}\left(G_f + \gamma G_g\right)^{3/2}}{\lambda^{3/2}\epsilon^{3/4}}\right. \\
& \left. + \frac{m^2\left(\ln m\right)^{2/3}\left(G_f + \gamma G_g + \gamma L_g\right)^{4/3}}{\lambda^{4/3}\epsilon^{2/3}} + \frac{m^3\left(\ln m\right)\left(G_f + \gamma G_g\right)}{\lambda\sqrt{\epsilon}}\right),
\end{aligned}$$

Algorithm 3 (MidTouch) Minimizes a λ-strongly convex f on $\mathcal{W}$ subject to the constraints $g_i(w) \leq 0$ for $i \in \{1, \dots, m\}$. The algorithm consists of two phases: the first T_1 iterations proceed like FullTouch, with every constraint being checked; the final T_2 iterations proceed like LightTouch, with only a constant number of constraints being checked during each iteration, and an auxiliary probability distribution p being learned along the way. Notice that while second-phase p-update checks only one constraint, it, like LightTouch, requires $O(m)$ arithmetic operations. This issue is discussed further in Section 5.1.

```
Hyperparameters: $T_1, T_2, \eta$
    // First phase
    Initialize $w^{(1)} \in \mathcal{W}$ arbitrarily
    For $t = 1$ to $T_1$:
        Sample $\check{\Delta}^{(t)}$        // stochastic subgradient of $f(w^{(t)})$
        Let $\check{\Delta}_w^{(t)} = \check{\Delta}^{(t)} + \gamma \check{\nabla} \max\{0, g(w^{(t)})\}$
        Update $w^{(t+1)} = \Pi_w(w^{(t)} - (1/\lambda t)\check{\Delta}_w^{(t)})$        // $\Pi_w$ projects its argument onto $\mathcal{W}$ w.r.t. $\|\cdot\|_2$
    // Second phase
    Average $w^{(T_1+1)} = (\sum_{t=1}^{T_1} w^{(t)})/T_1$        // initialize second phase to result of first
    Initialize $p^{(T_1+1)} \in \Delta^m$ to the uniform distribution
    Initialize $\mu_j^{(T_1+1)} = \max\{0, g_j(w^{(T_1+1)})\}$
    For $t = T_1 + 1$ to $T_1 + T_2$:
        Sample $\check{\Delta}^{(t)}$
        Sample $i^{(t)} \sim p^{(t)}$
        Let $\check{\Delta}_w^{(t)} = \check{\Delta}^{(t)} + \gamma \check{\nabla} \max\{0, g_{i^{(t)}}(w^{(t)})\}$
        Update $w^{(t+1)} = \Pi_w(w^{(t)} - (1/\lambda t)\check{\Delta}_w^{(t)})$
        Sample $j^{(t)} \sim$ Unif$\{1, \dots, m\}$
        Let $\hat{\Delta}_p^{(t)} = \gamma \mu^{(t)} + \gamma m e_{j^{(t)}} (\max\{0, g_{j^{(t)}}(w^{(t)})\} - \mu_{j^{(t)}}^{(t)})$
        Let $\mu_k^{(t+1)} = \mu_k^{(t)}$ if $k \neq j^{(t)}$, otherwise $\mu_{j^{(t)}}^{(t+1)} = \max\{0, g_{j^{(t)}}(w^{(t)})\}$
        Update $\tilde{p}^{(t+1)} = \exp(\ln p^{(t)} + \eta \hat{\Delta}_p^{(t)})$        // element-wise exp and ln
        Project $p^{(t+1)} = \tilde{p}^{(t+1)} / \left\|\tilde{p}^{(t+1)}\right\|_1$
    Average $\bar{w} = (\sum_{t=T_1+1}^{T_1+T_2} w^{(t)})/T_2$
    Return $\Pi_g(\bar{w})$        // optional if small constraint violations are acceptable
```

then:

$$\mathbb{E}\left[\left\|\Pi_g(\bar{w}) - w^*\right\|_2^2\right] \leq \mathbb{E}\left[\left\|\bar{w} - w^*\right\|_2^2\right] \leq \epsilon,$$

where $w^* = \operatorname{argmin}_{\{w \in \mathcal{W} : \forall i . g_i(w) \leq 0\}} f(w)$ *is the* optimal *constraint-satisfying reference vector.*

Proof In Appendix D. ■

Notice that the above theorem bounds not the suboptimality of $\Pi_g(\bar{w})$, but rather its squared Euclidean distance from w^*, for which reason the denominator of the highest order term depends on λ^2 rather than λ. Like Theorem 3 in the non-strongly convex case, the dominant terms above, both in terms of the total number of iterations and number of constraint checks, match the usual $1/\epsilon$ convergence rate for unconstrained strongly-convex SGD with an additional $\gamma^2 \ln m$ factor, while the lower-order terms have a worse m-dependence. As before, fewer constraint checks will be per-

Table 2: Comparison of the number of iterations, and number of constraint checks, required to achieve ϵ-suboptimality with high probability when optimizing a non-strongly-convex objective, up to constant and logarithmic factors, dropping the L_g, G_f and G_g dependencies, and ignoring the one-time cost of projecting the final result in FullTouch and LightTouch. For LLO-FW, the parameter to the local linear oracle has magnitude $O(\sqrt{d}\nu)$. See Section 4, Appendix C, the non-smooth stochastic result of Hazan and Kale (2012, Theorem 4.3), and Garber and Hazan (2013, Theorem 2). Notice that because this table compares upper bounds to upper bounds, subsequent work may improve these bounds further.

	#Iterations to achieve ϵ-suboptimality	**#Constraint checks to achieve ϵ-suboptimality**
FullTouch	$\frac{\gamma^2 D_w^2}{\epsilon^2}$	$\frac{m\gamma^2 D_w^2}{\epsilon^2}$
LightTouch	$\frac{(\ln m)\gamma^2 D_w^4}{\epsilon^2}$	$\frac{(\ln m)\gamma^2 D_w^4}{\epsilon^2} + \frac{m(\ln m)^{3/2}\gamma^{3/2} D_w^3}{\epsilon^{3/2}}$
Projected SGD	$\frac{D_w^2}{\epsilon^2}$	N/A (projection)
Online Frank-Wolfe	$\frac{D_w^3}{\epsilon^3}$	N/A (linear optimization)
LLO-FW	$\frac{d\nu^2 D_w^2}{\epsilon^2}$	N/A (local linear oracle)

formed per iteration as ϵ shrinks, reaching a constant number (on average) once ϵ is on the order of $1/m^6$.

4. Theoretical Comparison

Table 2 compares upper bounds on the convergence rates and per-iteration costs when applied to a convex (but not necessarily strongly convex) problem for LightTouch, FullTouch, projected SGD, the online Frank-Wolfe algorithm of Hazan and Kale (2012), and a Frank-Wolfe-like online algorithm for optimization over a polytope (Garber and Hazan, 2013). The latter algorithm, which we refer to as LLO-FW, achieves convergence rates comparable to projected SGD, but uses a local linear oracle instead of a projection or full linear optimization. To simplify the presentation, the dependencies on L_g, G_f and G_g have been dropped—please refer to Theorems 3 and 4 and the cited references for the complete statements. Table 3 contains the same comparison (without online Frank-Wolfe) for λ-strongly convex problems.

At each iteration, all of these algorithms must find a stochastic subgradient of f. In addition, each iteration of LightTouch and MidTouch must perform $O(m)$ arithmetic operations (for the m-dimensional vector operations used when updating p)—this issue will be discussed further in Section 5.1. However, projected SGD must project its iterate onto the constraints w.r.t. the Euclidean norm, online Frank-Wolfe must perform a linear optimization subject to the constraints, and LLO-FW must evaluate a local linear oracle, which amounts to essentially local linear optimization.

LightTouch, MidTouch and FullTouch share the same γ-dependence, but the m-dependence of the convergence rate of LightTouch and MidTouch is logarithmically worse. The number of constraint evaluations, however, is better: in the non-strongly convex case, ignoring all but the m and ϵ dependencies, FullTouch will check $O(m/\epsilon^2)$ constraints, while LightTouch will check only

Table 3: Same as Table 2, except that the results bound the number of iterations or constraint checks required to achieve $\mathbb{E}[\|w - w^*\|_2^2] \leq \epsilon$, and the objective function is assumed to be λ-strongly convex. The bound given for FullTouch assumes that the constant η used in Algorithm 1 has been replaced with the standard decreasing $1/\lambda t$ step size used in strongly-convex SGD. The MidTouch bounds each contain four terms, listed in order of most-to-least dominant (in ϵ). For LLO-FW, the parameter to the local linear oracle has magnitude $O(\sqrt{d}\nu)$. See Section 4, Appendix D, and Garber and Hazan (2013, Theorem 3).

	#Iterations to achieve ϵ-suboptimality
FullTouch	$\frac{\gamma^2 D_w^2}{\lambda^2 \epsilon}$
MidTouch	$\frac{(\ln m)\gamma^2 D_w^2}{\lambda^2 \epsilon} + \frac{m^{3/2} (\ln m)^{3/2} \gamma^{3/2} D_w^{3/2}}{\lambda^{3/2} \epsilon^{3/4}} + \frac{m (\ln m)^{2/3} \gamma^{4/3} D_w^{4/3}}{\lambda^{4/3} \epsilon^{2/3}} + \frac{m^2 (\ln m) \gamma D_w}{\lambda \sqrt{\epsilon}}$
Projected SGD	$\frac{D_w^2}{\lambda^2 \epsilon}$
LLO-FW	$\frac{d \nu^2 D_w^2}{\lambda^2 \epsilon}$
	#Constraint checks to achieve ϵ-suboptimality
FullTouch	$\frac{m \gamma^2 D_w^2}{\lambda^2 \epsilon}$
MidTouch	$\frac{(\ln m)\gamma^2 D_w^2}{\lambda^2 \epsilon} + \frac{m^{3/2} (\ln m)^{3/2} \gamma^{3/2} D_w^{3/2}}{\lambda^{3/2} \epsilon^{3/4}} + \frac{m^2 (\ln m)^{2/3} \gamma^{4/3} D_w^{4/3}}{\lambda^{4/3} \epsilon^{2/3}} + \frac{m^3 (\ln m) \gamma D_w}{\lambda \sqrt{\epsilon}}$
Projected SGD	N/A (projection)
LLO-FW	N/A (local linear oracle)

$\tilde{O}((\ln m)/\epsilon^2 + m/\epsilon^{3/2})$, a significant improvement when ϵ is small. Hence, particularly for problems with many expensive-to-evaluate constraints, one would expect LightTouch to converge much more rapidly. Likewise, for λ-strongly convex optimization, the dominant (in ϵ) terms in the bounds on the number of constraint evaluations go as m/ϵ for FullTouch, and as $(\ln m)/\epsilon$ for MidTouch, although the lower-order terms in the MidTouch bound are significantly more complex than in the non-strongly convex case (see Table 3 for full details).

Comparing with projected SGD, online Frank-Wolfe and LLO-FW is less straightforward, not only because we're comparing upper bounds to upper bounds (with all of the uncertainty that this entails), but also because we must relate the value of γ to the cost of performing the required projection, constrained linear optimization or local linear oracle evaluation. We note, however, that for non-strongly convex optimization, the ϵ-dependence of the convergence rate bound is worse for online Frank-Wolfe ($1/\epsilon^3$) than for the other algorithms ($1/\epsilon^2$), and that unless the constraints have some special structure, performing a projection can be a very expensive operation.

For example, with general linear inequality constraints, each constraint check performed by LightTouch, MidTouch or FullTouch requires $O(d)$ time, whereas each linear program optimized by online Frank-Wolfe could be solved in $O(d^2 m)$ time (Nemirovski, 2004, Chapter 10.1), and each projection performed by SGD in $O((dm)^{3/2})$ time (Goldfarb and Liu, 1991). When the constraints are taken to be arbitrary convex functions, instead of linear functions, projections may be even more difficult.

Algorithm 4 (Practical LightTouch) Our proposed "practical" algorithm combining LightTouch and MidTouch, along with the changes discussed in Section 5.

```
Hyperparameters: $T$, $\eta_w$, $\eta_p$
  Initialize $w^{(1)} \in \mathcal{W}$ arbitrarily
  Initialize $p^{(1)} \in \Delta^m$ to the uniform distribution
  Initialize $\mu_j^{(1)} = \max\{0, g_j(w^{(1)})\}$        // 0 if $w^{(1)}$ is feasible
  For $t = 1$ to $T$:
      Let $\eta_w^{(t)} = \eta_w / t$ if $f$ is strongly convex, $\eta_w / \sqrt{t}$ otherwise
      Set $k_f^{(t)}$, $k_g^{(t)}$ and $k_p^{(t)}$ as described in Section 5.2
      Sample $\check{\Delta}_1^{(t)}, \dots, \check{\Delta}_{k_f^{(t)}}^{(t)}$ i.i.d.        // stochastic subgradients of $f(w^{(t)})$
      Sample $i_1^{(t)}, \dots, i_{k_g^{(t)}}^{(t)} \sim p^{(t)}$ i.i.d.
      Let $\check{\Delta}_w^{(t)} = (1/k_f^{(t)}) \sum_{j=1}^{k_f^{(t)}} \check{\Delta}_j^{(t)} + (\gamma/k_g^{(t)}) \sum_{j=1}^{k_g^{(t)}} \check{\nabla} \max\{0, g_{i_j^{(t)}}(w^{(t)})\}$
      Update $w^{(t+1)} = \Pi_w(w^{(t)} - \eta_w^{(t)} \check{\Delta}_w^{(t)})$        // $\Pi_w$ projects its argument onto $\mathcal{W}$ w.r.t. $\|\cdot\|_2$
      Sample $S^{(t)} \subseteq \{1, \dots, m\}$ with $|S^{(t)}| = k_p^{(t)}$ uniformly without replacement
      Let $\hat{\Delta}_p^{(t)} = \gamma \mu^{(t)} + (\gamma m / k_p^{(t)}) \sum_{j \in S^{(t)}} e_j (\max\{0, g_j(w^{(t)})\} - \mu_j^{(t)})$
      Let $\mu_j^{(t+1)} = \max\{0, g_j(w^{(t)})\}$ if $j \in S^{(t)}$, otherwise $\mu_j^{(t+1)} = \mu_j^{(t)}$
      Update $\tilde{p}^{(t+1)} = \exp(\ln p^{(t)} + \eta_p \hat{\Delta}_p^{(t)})$        // element-wise $\exp$ and $\ln$
      Project $p^{(t+1)} = \tilde{p}^{(t+1)} / \|\tilde{p}^{(t+1)}\|_1$
  Average $\bar{w} = (\sum_{t=1}^{T} w^{(t)}) / T$
  Return $\Pi_g(\bar{w})$        // optional if small constraint violations are acceptable
```

We believe that in many cases γ^2 will be roughly on the order of the dimension d, or number of constraints m, whichever is smaller, although it can be worse for difficult constraint sets (see Section 2.2). In practice, we have found that a surprisingly small γ—we use $\gamma = 1$ in our experiments (Section 6)—often suffices to result in convergence to a feasible solution. With this in mind, and in light of the fact that a fast projection, linear optimization, or local linear oracle evaluation may only be possible for particular constraint sets, we believe that our algorithms compare favorably with the alternatives.

5. Practical Considerations

Algorithms 2 and 3 were designed primarily to be easy to analyze, but in real world-applications we recommend making a few tweaks to improve performance. The first of these is trivial: using a decreasing w-update step size $\eta_w^{(t)} = \eta_w / \sqrt{t}$ when optimizing a non-strongly convex objective, and $\eta_w^{(t)} = \eta_w / t$ for a strongly-convex objective. In both cases we continue to use a constant p-update step size η_p. This change, as well as that described in Section 5.2, is included in Algorithm 4.

5.1. Constraint Aggregation

A natural concern about Algorithms 2 and 3 is that $O(m)$ arithmetic operations are performed per iteration, even when only a few constraints are checked. When each constraint is expensive, this is a minor issue, since this cost will be "drowned out" by that of checking the constraints. However, when the constraints are very cheap, and the $O(m)$ arithmetic cost compares disfavorably with the cost of checking a handful of constraints, it can become a bottleneck.

Our solution to this issue is simple: transform a problem with a large number of cheap constraints into one with a smaller number of more expensive constraints. To this end, we partition the constraint indices $1, \dots, m$ into $\tilde{m}$ sets $\{M_i\}$ of size at most $\lceil m/\tilde{m} \rceil$, defining $\tilde{g}_i(w) = \max_{j \in M_i} g_j(w)$, and then apply LightTouch or MidTouch on the $\tilde{m}$ aggregated constraints $\tilde{g}_i(w) \leq 0$. This makes each constraint check $\lceil m/\tilde{m} \rceil$ times more expensive, but reduces the dimension of p from m to $\tilde{m}$, shrinking the per-iteration arithmetic cost to $O(\tilde{m})$.

5.2. Automatic Minibatching

Because LightTouch takes a minibatch size k as a parameter, and the constants from which we derive the recommended choice of k (Theorem 3) are often unknown, a user is in the uncomfortable position of having to perform a parameter search not only over the step sizes η_w and η_p, but also the minibatch size. Furthermore, the fact that the theoretically-recommended k is a decreasing function of T indicates that it might be better to check more constraints in early iterations, and fewer in later ones. Likewise, MidTouch is structured as a two-phase algorithm, in which every iteration checks every constraint in the first phase, and only a constant number in the second, but it seems more sensible for the number of constraint checks to decrease *gradually* over time.

In addition, for both algorithms, it would be desirable to support separate minibatching of the loss and constraint stochastic subgradients (w.r.t. w), in which case there would be three minibatching parameters to determine: k_f, k_g and k_p. This makes things even harder for the user, since now there are *three* additional parameters that must be specified.

To remove the need to specify any minibatch-size hyperparameters, and to enable the minibatch sizes to change from iteration-to-iteration, we propose a heuristic that will automatically determine the minibatch sizes $k_f^{(t)}$, $k_g^{(t)}$ and $k_p^{(t)}$ for each of the stochastic gradient components at each iteration. Intuitively, we want to choose minibatch sizes in such a way that the stochastic gradients are both cheap to compute and have low variance. Our proposed heuristic does this by trading-off the computational cost and "bound impact" of the overall stochastic gradient, where the "bound impact" is a variance-like quantity that approximates the impact that taking a step with particular minibatch sizes has on the relevant convergence rate bound.

Suppose that we're about to perform the tth iteration, and know that a *single* stochastic subgradient $\check{\Delta}$ of $f(w)$ (corresponding to the loss portion of $\check{\Delta}_w$) has variance (more properly, covariance matrix trace) $\bar{v}_f^{(t)}$ and requires a computational investment of $\bar{c}_f^{(t)}$ units. Similarly, if we define $\check{\Delta}_g$ by sampling $i \sim p$ and taking $\check{\Delta}_g = \gamma \check{\nabla} \max\{0, g_i(w)\}$ (corresponding to the constraint portion of $\check{\Delta}_w$), then we can define variance and cost estimates of $\check{\Delta}_g$ to be $\bar{v}_g^{(t)}$ and $\bar{c}_g^{(t)}$, respectively. Likewise, we take $\bar{v}_p^{(t)}$ and $\bar{c}_p^{(t)}$ to be estimates of the variance and cost of a (non-minibatched version of) $\hat{\Delta}_p$.

In all three cases, the variance and cost estimates are those of a *single sample*, implying that a stochastic subgradient of $f(w)$ averaged over a minibatch of size $k_f^{(t)}$ will have variance $\bar{v}_f^{(t)}/k_f^{(t)}$ and require a computational investment of $\bar{c}_f^{(t)} k_f^{(t)}$, and likewise for the constraints and distribution.

In the context of Algorithm 4, with minibatch sizes of $k_f^{(t)}$, $k_g^{(t)}$ and $k_p^{(t)}$, we define the overall bound impact b and computational cost c of a single update as:

$$b = \frac{\eta_w^{(t)} \bar{v}_f^{(t)}}{k_f^{(t)}} + \frac{\eta_w^{(t)} \bar{v}_g^{(t)}}{k_g^{(t)}} + \frac{\eta_p \bar{v}_p^{(t)}}{k_p^{(t)}} \qquad \text{and} \qquad c = \bar{c}_f^{(t)} k_f^{(t)} + \bar{c}_g^{(t)} k_g^{(t)} + \bar{c}_p^{(t)} k_p^{(t)}.$$

We should emphasize that the above definition of b is merely a useful approximation of how these quantities truly affect our bounds.

Given the three variance and three cost estimates, we choose minibatch sizes in such a way as to minimize both the computational cost and bound impact of an update. Imagine that we are given a fixed computational budget c. Then our goal will be to choose the minibatch sizes in such a way that b is minimized for this budget, a problem that is easily solved in closed form:

$$\left[k_f^{(t)}, k_g^{(t)}, k_p^{(t)}\right] \propto \left[\sqrt{\frac{\eta_w^{(t)} \bar{v}_f^{(t)}}{\bar{c}_f^{(t)}}}, \sqrt{\frac{\eta_w^{(t)} \bar{v}_g^{(t)}}{\bar{c}_g^{(t)}}}, \sqrt{\frac{\eta_p \bar{v}_p^{(t)}}{\bar{c}_p^{(t)}}}\right].$$

We propose choosing the proportionality constant (and thereby the cost budget c) in such a way that $k_f^{(t)} = 2$ (enabling us to calculate sample variances, as explained below), and round the two other sizes to the nearest integers, lower-bounding each so that $k_g^{(t)} \geq 2$ and $k_p^{(t)} \geq 1$.

While the variances and costs are not truly *known* during optimization, they are easy to estimate from known quantities. For the costs $\bar{c}_f^{(t)}$, $\bar{c}_g^{(t)}$ and $\bar{c}_p^{(t)}$, we simply time how long each past stochastic gradient calculation has taken, and then average them to estimate the future costs. For the variances $\bar{v}_f^{(t)}$ and $\bar{v}_g^{(t)}$, we restrict ourselves to minibatch sizes $k_f^{(t)}, k_g^{(t)} \geq 2$, calculate the sample variances $v_f^{(t)}$ and $v_g^{(t)}$ of the stochastic gradients at each iteration, and then average over all past iterations (either uniformly, or a weighted average placing more weight on recent iterations).

For $\bar{v}_p^{(t)}$, the situation is a bit more complicated, since the p-updates are multiplicative (so we should use an ℓ^∞ variance) and centered as in Equation 4. Upper-bounding the ℓ^∞ norm with the ℓ^2 norm and using the fact that the minibatch $S^{(t)}$ is independently sampled yields the following crude estimate:

$$v_p^{(t)} = \gamma^2 m^2 \left(\frac{1}{k_p^{(t)}} \sum_{i \in S^{(t)}} \left(\mu_i - \max\left\{0, g_i\left(w^{(t)}\right)\right\}\right)^2 \right),$$

We again average $v_p^{(t)}$ across past iterations to estimate $\bar{v}_p^{(t)}$.

6. Experiments

We validated the performance of our practical variant of LightTouch (Algorithm 4) on a YouTube ranking problem in the style of Joachims (2002), in which the task is to predict what a user will watch next, given that they have just viewed a certain video. In this setting, a user has just viewed video a, was presented with a list of candidate videos to watch next, and clicked on b^+, with b^-

being the video immediately preceding b^+ in the list (if b^+ was the first list element, then the example is thrown out).

We used an anonymized proprietary dataset consisting of $n = 612\,587$ training pairs of feature vectors (x^+, x^-), where x^+ is a vector of 12 features summarizing the similarity between a and b^+, and x^- between a and b^-.

We treat this as a standard pairwise ranking problem, for which the goal is to estimate a function $f(\Phi(x)) = \langle w, \Phi(x) \rangle$ such that $f(\Phi(x^+)) > f(\Phi(x^-))$ for as many examples as possible, subject to the appropriate regularization (or, in this case, constraints). Specifically, the (unconstrained) learning task is to minimize the average empirical hinge loss:

$$\min_{w \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \left(\max \left\{ 0, 1 - \left\langle w, \Phi\left(x_i^+\right) - \Phi\left(x_i^-\right) \right\rangle \right\} \right).$$

All twelve of the features were designed to provide positive evidence—in other words, if any one increases (holding the others fixed), then we expect $f(\Phi(x))$ to increase. We have found that using constraints to enforce this monotonicity property results in a better model in practice.

We define $\Phi(\cdot)$ as in lattice regression using simplex interpolation (Garcia et al., 2012; Gupta et al., 2016), an approach which works well at combining a small number of informative features, and more importantly (for our purposes) enables one to force the learned function to be monotonic via linear inequality constraints on the parameters. For the resulting problem, the feature vectors have dimension $d = 2^{12} = 4096$, we chose $\mathcal{W}$ to be defined by the box constraints $-10 \leq w_i \leq 10$ in each of the 4096 dimensions, and the total number of monotonicity-enforcing linear inequality constraints is $m = 24\,576$.

Every $\Phi(x)$ contains only $d + 1 = 13$ nonzeros and can be computed in $O(d \ln d)$ time. Hence, stochastic gradients of f are inexpensive to compute. Likewise, checking a monotonicity constraint only requires a single comparison between two parameter values, so although there are a large number of them, each constraint is very inexpensive to check.

6.1. Implementations

We implemented all algorithms in C++. Before running our main experiments, we performed crude parameter searches on a power-of-four grid (i.e. $\ldots, 1/16, 1/4, 1, 4, 16, \ldots$). For each candidate value we performed roughly $10\,000$ iterations, and chose the parameter that appeared to result in the fastest convergence in terms of the objective function.

LightTouch Our implementation of LightTouch includes all of the suggested changes of Section 5, including the constraint aggregation approach of Section 5.1, although we used no aggregation until our timing comparison (Section 6.3). For automatic minibatching, we took weighted averages of the variance estimates as $\bar{v}^{(t+1)} \propto v^{(t)} + \nu \bar{v}^{(t)}$. We found that up-weighting recent estimates (taking $\nu < 1$) resulted in a noticeable improvement, but that the precise value of ν mattered little (we used $\nu = 0.999$). Based on the grid search described above, we chose $\gamma = 1$, $\eta_w = 16$ and $\eta_p = 1/16$.

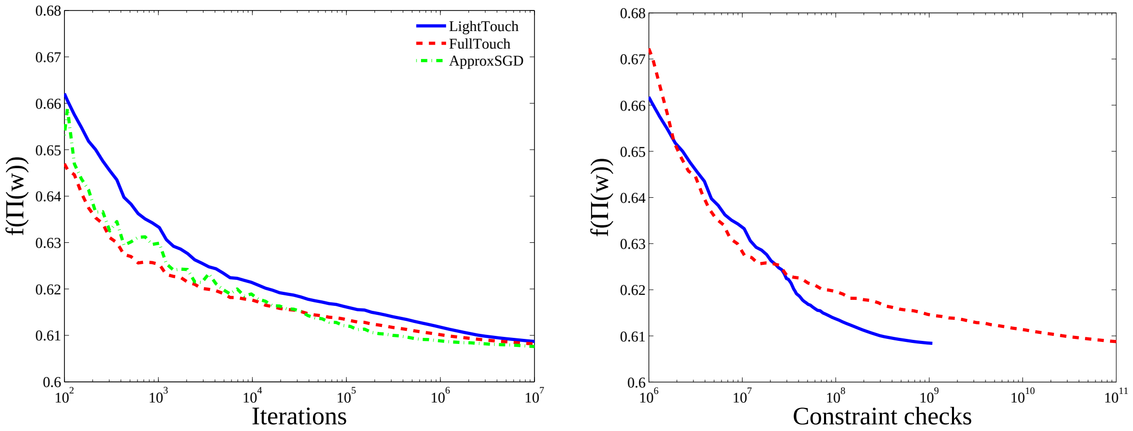

Figure 1: Comparison of convergence rates of LightTouch, FullTouch and ApproxSGD on the YouTube ranking problem of Section 6. The two plots show the objective function (average training hinge loss) $f(\Pi_g(w^{(t)}))$ as a function of the number of iterations, and as a function of the total number of times a single constraint function g_i was evaluated or differentiated, respectively.

FullTouch Our FullTouch implementation differs from that in Algorithm 1 only in that we used a decreasing step size $\eta_w^{(t)} = \eta_w/\sqrt{t}$. As with LightTouch, we chose $\gamma = 1$ and $\eta_w = 16$ based on a grid search.

ProjectedSGD We implemented Euclidean projections onto lattice monotonicity constraints using IPOPT (Wächter and Biegler, 2006) to optimize the resulting sparse 4096-dimensional quadratic program. However, the use of a QP solver for projected SGD—a very heavyweight solution—resulted in an implementation that was too slow to experiment with, requiring nearly four minutes per projection (observe that our experiments each ran for millions of iterations).

ApproxSGD This is an approximate projected SGD implementation using the fast approximate update procedure described in Gupta et al. (2016), which is an active set method that, starting from the current iterate, moves along the boundary of the feasible region, adding constraints to the active set as they are encountered, until the desired step is exhausted (this is reminiscent of the local linear oracles considered by Garber and Hazan (2013)). This approach is particularly well-suited to this particular constraint set because (1) when checking constraints for possible inclusion in the active set, it exploits the sparsity of the stochastic gradients to only consider monotonicity constraints which could possibly be violated, and (2) projecting onto an intersection of active monotonicity constraints reduces to uniformly averaging every set of parameters that are "linked together" by active constraints. Like the other algorithms, we used step sizes of $\eta_w^{(t)} = \eta_w/\sqrt{t}$ and chose $\eta_w = 64$ based on the grid search (recall that $\eta_w = 16$ was better for the other two algorithms).

In every experiment we repeatedly looped over a random permutation of the training set, and generated plots by averaging over 5 such runs (with the same 5 random permutations) for each algorithm.

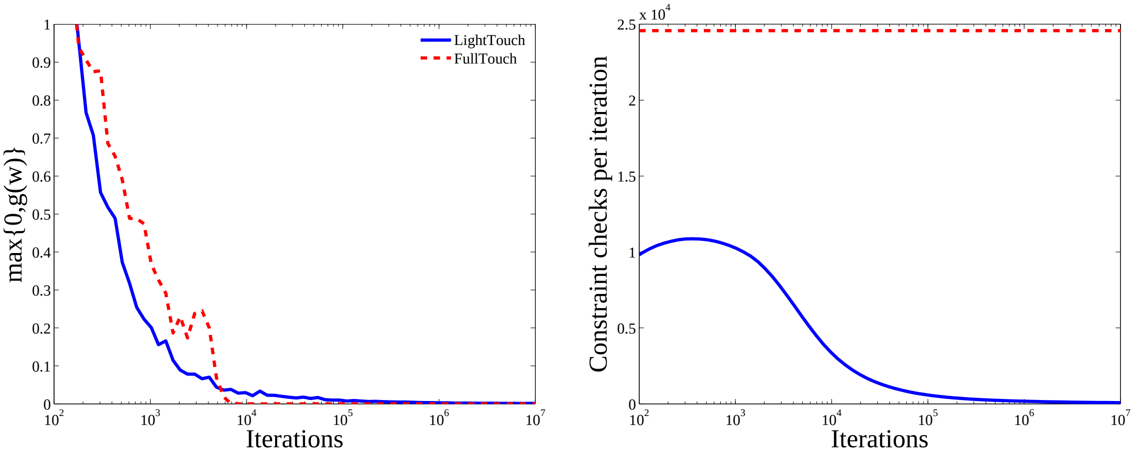

Figure 2: Comparison constraint-handling of LightTouch and FullTouch on the YouTube ranking problem of Section 6. The two plots show the constraint violation magnitude $\max\{0, g(w^{(t)})\}$, and the average number of constraints checked per iteration up to this point, respectively, both as functions of the number of iterations.

6.2. Constraint-check Comparison

In our first set of experiments, we compared the performance of LightTouch, FullTouch and ApproxSGD in terms of the number of stochastic subgradients of f drawn, and the number of constraints checked. Because LightTouch's automatic minibatching fixes $k_f^{(t)} = 2$ (with the other two minibatch sizes being automatically determined), in these experiments we used minibatch sizes of 2 for FullTouch and ApproxSGD, guaranteeing that all three algorithms observe the same number of stochastic subgradients of f at each iteration.

The left-hand plot of Figure 1 shows that all three algorithms converge at roughly comparable per-iteration rates, with ApproxSGD having a slight advantage over FullTouch, which itself converges a bit more rapidly than LightTouch. The right-hand plot shows a striking difference, however—LightTouch reaches a near-optimal solution having checked more than $10\times$ fewer constraints than FullTouch. Notice that we plot the suboptimalities of the projected iterates $\Pi_w(w^{(t)})$ rather than of the $w^{(t)}$s themselves, in order to emulate the final projection (line 7 of Algorithm 1 and 17 of Algorithm 4), and guarantee that we only compare the average losses of *feasible* intermediate solutions.

In Figure 2, we explore how well our algorithms enforce feasibility, and how effective automatic minibatching is at choosing minibatch sizes. The left-hand plot shows that both FullTouch has converged to a nearly-feasible solution after roughly $10\,000$ iterations, and LightTouch (unsurprisingly) takes more, perhaps $100\,000$ or so. In the right-hand plot, we see that, in line with our expectations (see Section 5.2), LightTouch's automatic minibatching results in very few constraints being checked in late iterations.

6.3. Timing Comparison

Our final experiment compared the wall-clock runtimes of our implementations. Note that, because each monotonicity constraint can be checked with only a single comparison (compare with e.g. $O(d)$ arithmetic operations for a dense linear inequality constraint), the $O(m)$ arithmetic cost of maintaining and updating the probability distribution p over the constraints is significant. Hence, in terms of the constraint costs, this is nearly a worse-case problem for LightTouch. We experimented with power-of-4 constraint aggregate sizes (Section 5.1), and found that using $\tilde{m} = 96$ aggregated constraints, each of size 256, worked best.

FullTouch, without minibatching, draws a single stochastic subgradient of f and checks every constraint at each iteration. However, it would seem to be more efficient to use minibatching to look at more stochastic subgradients at each iteration, and therefore fewer constraints per stochastic subgradient of f. Hence, for FullTouch, we again searched over power-of-4 minibatch sizes, and found that 16 worked best.

For ApproxSGD, the situation is less clear-cut. On the one hand, increasing the minibatch size results in fewer approximate projections being performed per stochastic subgradient of f. On the other, averaging more stochastic subgradients results in less sparsity, slowing down the approximate projection. We found that the latter consideration wins out—after searching again over power-of-4 minibatch sizes, we found that a minibatch size of 1 (i.e. no minibatching) worked best.

Figure 3 contains the results of these experiments, showing that both FullTouch and LightTouch converge significantly faster than ApproxSGD. Interestingly, ApproxSGD is rather slow in early iterations (clipped off in plot), but accelerates in later iterations. We speculate that the reason for this behavior is that, close to convergence, the steps taken at each iteration are smaller, and therefore the active sets constructed during the approximate projection routine do not grow as large. FullTouch enjoys a small advantage over LightTouch until both algorithms are very close to convergence, but based on the results of Section 6.2, we believe that this advantage would reverse if there were more constraints, or if the constraints were more expensive to check.

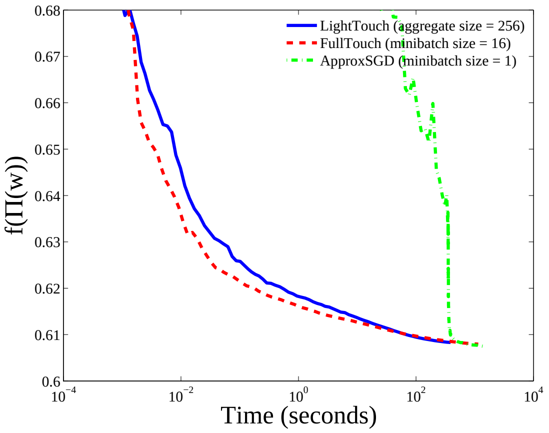

Figure 3: Plot of the objective function (average training hinge loss) $f(\Pi_g(w^{(t)}))$ as a function of runtime for our implementations of LightTouch, FullTouch and ApproxSGD, on the YouTube ranking problem of Section 6.

7. Conclusions

We have proposed an efficient strategy for large-scale heavily constrained optimization, building on the work of Mahdavi et al. (2012), and analyze its performance, demonstrating that, asymptotically, our approach requires many fewer constraint checks in order to converge.

We build on these theoretical results to propose a practical variant. The most significant of these improvements is based on the observation that our algorithm takes steps based on three separate stochastic gradients, and that trading off the variances of computational costs of these three components is beneficial. To this end, we propose a heuristic for dynamically choosing minibatch sizes in such a way as to encourage faster convergence at a lower computational cost.

Experiments on a real-world 4096-dimensional machine learning problem with $24\,576$ constraints and $612\,587$ training examples—too large for a QP-based implementation of projected SGD—showed that our proposed method is effective. In particular, we find that, in practice, our technique checks fewer constraints per iteration than competing algorithms, and, as expected, checks ever fewer as optimization progresses.

Acknowledgments

We thank Kevin Canini, Mahdi Milani Fard, Andrew Frigyik, Michael Friedlander and Seungil You for helpful discussions, and proofreading earlier drafts.

References

N. P. Archer and S. Wang. Application of the back propagation neural network algorithm with monotonicity constraints for two-group classification problems. *Decision Sciences*, 24:60–75, 1993.

H. H. Bauschke. *Projection Algorithms and Monotone Operators*. Ph.D. Thesis, Simon Fraser University, Canada, 1996.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, May 2003.

K. L. Clarkson, E. Hazan, and D. P. Woodruff. Sublinear optimization for machine learning. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS'10, pages 449–457, Washington, DC, USA, 2010. IEEE Computer Society.

H. Daniels and M. Velikova. Monotone and partially monotone neural networks. *IEEE Trans. Neural Networks*, 21(6):906–917, 2010.

K. Dzhaparidze and J. H. van Zanten. On Bernstein-type inequalities for martingales. *Stochastic Processes and their Applications*, 93(1):109–117, May 2001.

D. Garber and E. Hazan. Playing non-linear games with linear oracles. In *FOCS*, pages 420–428. IEEE Computer Society, 2013.

E. K. Garcia, R. Arora, and M. Gupta. Optimized regression for efficient function evaluation. *IEEE Trans. Image Processing*, 21(9):4128–4140, September 2012.

D. Goldfarb and S. Liu. An O(Ln^3) primal interior point algorithm for convex quadratic programming. *Math. Program.*, 49(3):325–340, January 1991.

M. R. Gupta, A. Cotter, J. Pfeifer, K. Voevodski, K. Canini, A. Mangylov, W. Moczydlowski, and A. van Esbroeck. Monotonic calibrated interpolated look-up tables. *JMLR (to appear)*, 2016.

E. Hazan and S. Kale. Projection-free online learning. In *ICML'12*, 2012.

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML'13*, volume 28, pages 427–435, 2013.

T. Joachims. Optimizing search engines using clickthrough data. In *KDD'02*, pages 133–142, New York, NY, USA, 2002.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS'13*, pages 315–323. 2013.

M. Mahdavi, T. Yang, R. Jin, S. Zhu, and J. Yi. Stochastic gradient descent with only one projection. In *NIPS'12*, pages 494–502. 2012.

A. Nemirovski. Lecture notes: Interior point polynomial time methods in convex programming. 2004. URL `http://www2.isye.gatech.edu/~nemirovs/Lect_IPM.pdf`.

A. Nemirovski and D. Yudin. *Problem complexity and method efficiency in optimization*. John Wiley & Sons Ltd, 1983.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, January 2009.

Y.-J. Qu and B.-G. Hu. Generalized constraint neural network regression model subject to linear priors. *IEEE Trans. on Neural Networks*, 22(11):2447–2459, 2011.

A. Rakhlin and K. Sridharan. Optimization, learning, and games with predictable sequences. In *NIPS'13*, pages 3066–3074. 2013.

S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. *Mathematical Programming*, 127(1):3–30, March 2011.

J. Sill. Monotonic networks. *Advances in Neural Information Processing Systems (NIPS)*, 1998.

J. Spouge, H. Wan, and W. J. Wilbur. Least squares isotonic regression in two dimensions. *Journal of Optimization Theory and Applications*, 117(3):585–605, 2003.

N. Srebro, K. Sridharan, and A. Tewari. On the universality of online mirror descent. In *NIPS'11*, 2011.

A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1): 25–57, 2006.

M. Wang, Y. Chen, J. Liu, and Y. Gu. Random Multi-Constraint Projection: Stochastic Gradient Methods for Convex Optimization with Many Constraints. *ArXiv e-prints*, November 2015.

Wikipedia. Coupon collector's problem — Wikipedia, the free encyclopedia, 2014. URL `http://en.wikipedia.org/wiki/Coupon_collector%27s_problem`. [Online; accessed 20-November-2014].

Wikipedia. Properties of polynomial roots — Wikipedia, the free encyclopedia, 2015. URL `http://en.wikipedia.org/wiki/Properties_of_polynomial_roots`. [Online; accessed 27-March-2015].

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML'03*, 2003.

Table 4: New notation in Appendix A.

Symbol	Description	Definition
$\mathcal{A}$	Convex domain of dual variables	
$\mathcal{F}$	Filtration	$\check{\Delta}^{(t)}$, $\check{\Delta}_w^{(t)}$, $\hat{\Delta}_\alpha^{(t)}$ are $\mathcal{F}_t$-measurable
$\left\|\cdot\right\|$, $\left\|\cdot\right\|_*$	Unspecified norm and its dual	
$\left\|\cdot\right\|_w$, $\left\|\cdot\right\|_{w*}$	Norm on $\mathcal{W}$ and its dual	
$\left\|\cdot\right\|_\alpha$, $\left\|\cdot\right\|_{\alpha*}$	Norm on $\mathcal{A}$ and its dual	
Ψ, Ψ^*	A d.g.f. and its convex conjugate	
Ψ_w, Ψ_w^*	A d.g.f. on $\mathcal{W}$ and its convex conjugate	
$\Psi_\alpha, \Psi_\alpha^*$	A d.g.f. on $\mathcal{A}$ and its convex conjugate	
$\check{\Delta}$	Stochastic subgradient	
$\check{\Delta}_w$	Primal stochastic subgradient	
$\hat{\Delta}_\alpha$	Dual stochastic supergradient	
R_*	Bound on w^*-centered radius of $\mathcal{W}$	$R_* \geq \left\|w - w^*\right\|$
R_{w*}	Bound on w^*-centered radius of $\mathcal{W}$	$R_{w*} \geq \left\|w - w^*\right\|_w$
$R_{\alpha*}$	Bound on α^*-centered radius of $\mathcal{A}$	$R_{\alpha*} \geq \left\|\alpha - \alpha^*\right\|_\alpha$
σ	Bound on $\check{\Delta}$ error	$\sigma \geq \left\|\mathbb{E}[\check{\Delta}^{(t)} \mid \mathcal{F}_{t-1}] - \check{\Delta}^{(t)}\right\|_*$
σ_w	Bound on $\check{\Delta}_w$ error	$\sigma_w \geq \left\|\mathbb{E}[\check{\Delta}_w^{(t)} \mid \mathcal{F}_{t-1}] - \check{\Delta}_w^{(t)}\right\|_{w*}$
σ_α	Bound on $\hat{\Delta}_\alpha$ error	$\sigma_\alpha \geq \left\|\mathbb{E}[\hat{\Delta}_\alpha^{(t)} \mid \mathcal{F}_{t-1}] - \hat{\Delta}_\alpha^{(t)}\right\|_{\alpha*}$
$1 - \delta_\sigma$	Probability that σ bound holds	
$1 - \delta_{\sigma w}$	Probability that σ_w bound holds	
$1 - \delta_{\sigma\alpha}$	Probability that σ_α bound holds	

Appendix A. Mirror Descent

Mirror descent (Nemirovski and Yudin, 1983; Beck and Teboulle, 2003) is a meta-algorithm for stochastic optimization (more generally, online regret minimization) which performs gradient updates with respect to a meta-parameter, the *distance generating function* (d.g.f.). The two most widely-used d.g.f.s are the squared Euclidean norm and negative Shannon entropy, for which the resulting MD instantiations are stochastic gradient descent (SGD) and a multiplicative updating algorithm, respectively. These are precisely the two d.g.f.s which our constrained algorithm will use for the updates of w and p. We'll here give a number of results which differ only slightly from "standard" ones, beginning with a statement of an online MD bound adapted from Srebro et al. (2011):

Theorem 5 *Let $\left\|\cdot\right\|$ and $\left\|\cdot\right\|_*$ be a norm and its dual. Suppose that the distance generating function (d.g.f.) Ψ is 1-strongly convex w.r.t. $\left\|\cdot\right\|$. Let Ψ^* be the convex conjugate of Ψ, and take $B_\Psi(w|w') = \Psi(w) - \Psi(w') - \langle \nabla\Psi(w'), w - w'\rangle$ to be the associated Bregman divergence.*

Take $f_t : \mathcal{W} \to \mathbb{R}$ to be a sequence of convex functions on which we perform T iterations of mirror descent starting from $w^{(1)} \in \mathcal{W}$:

$$\tilde{w}^{(t+1)} = \nabla\Psi^*\left(\nabla\Psi\left(w^{(t)}\right) - \eta\check{\nabla} f_t\left(w^{(t)}\right)\right),$$

$$w^{(t+1)} = \underset{w \in \mathcal{W}}{\operatorname{argmin}} B_\Psi\left(w \;\middle|\; \tilde{w}^{(t+1)}\right),$$

where $\check{\nabla} f_t(w^{(t)}) \in \partial f_t(w^{(t)})$ *is a subgradient of* f_t *at* $w^{(t)}$*. Then:*

$$\frac{1}{T}\sum_{t=1}^{T}\left(f_t\left(w^{(t)}\right) - f_t\left(w^*\right)\right) \leq \frac{B_\Psi\left(w^* \mid w^{(1)}\right)}{\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T}\left\|\check{\nabla} f_t\left(w^{(t)}\right)\right\|_*^2,$$

where $w^* \in \mathcal{W}$ *is an arbitrary reference vector.*

Proof This proof is essentially the same as that of Srebro et al. (2011, Lemma 2). By convexity:

$$\begin{aligned}\eta\left(f_t\left(w^{(t)}\right) - f_t\left(w^*\right)\right) &\leq \left\langle \eta\check{\nabla} f_t\left(w^{(t)}\right), w^{(t)} - w^*\right\rangle \\ &\leq \left\langle \eta\check{\nabla} f_t\left(w^{(t)}\right), w^{(t)} - \tilde{w}^{(t+1)}\right\rangle + \left\langle \eta\check{\nabla} f_t\left(w^{(t)}\right), \tilde{w}^{(t+1)} - w^*\right\rangle.\end{aligned}$$

By Hölder's inequality, $\langle w', w\rangle \leq \|w'\|\,\|w\|_*$. Also, $\Psi(w) = \sup_v(\langle v, w\rangle - \Psi^*(v))$ is maximized when $\nabla\Psi^*(v) = w$, so $\nabla\Psi(\nabla\Psi^*(v)) = v$. These results combined with the definition of $\tilde{w}^{(t+1)}$ give:

$$\begin{aligned}\eta\left(f_t\left(w^{(t)}\right) - f_t\left(w^*\right)\right) \leq{}& \left\|\eta\check{\nabla} f_t\left(w^{(t)}\right)\right\|_* \left\|w^{(t)} - \tilde{w}^{(t+1)}\right\| \\ &+ \left\langle \nabla\Psi\left(w^{(t)}\right) - \nabla\Psi\left(\tilde{w}^{(t+1)}\right), \tilde{w}^{(t+1)} - w^*\right\rangle.\end{aligned}$$

Using Young's inequality and the definition of the Bregman divergence:

$$\begin{aligned}\eta\left(f_t\left(w^{(t)}\right) - f_t\left(w^*\right)\right) \leq{}& \frac{1}{2}\left\|\eta\check{\nabla} f_t\left(w^{(t)}\right)\right\|_*^2 + \frac{1}{2}\left\|w^{(t)} - \tilde{w}^{(t+1)}\right\|^2 \\ &+ B_\Psi\left(w^* \,\middle|\, w^{(t)}\right) - B_\Psi\left(w^* \,\middle|\, \tilde{w}^{(t+1)}\right) - B_\Psi\left(\tilde{w}^{(t+1)} \,\middle|\, w^{(t)}\right).\end{aligned}$$

Applying the 1-strong convexity of Ψ to cancel the $\left\|w^{(t)} - \tilde{w}^{(t+1)}\right\|^2/2$ and $B_\Psi(\tilde{w}^{(t+1)} \mid w^{(t)})$ terms:

$$\eta\left(f_t\left(w^{(t)}\right) - f_t\left(w^*\right)\right) \leq \frac{\eta^2}{2}\left\|\check{\nabla} f_t\left(w^{(t)}\right)\right\|_*^2 + B_\Psi\left(w^* \,\middle|\, w^{(t)}\right) - B_\Psi\left(w^* \,\middle|\, \tilde{w}^{(t+1)}\right).$$

Summing over t, using the nonnegativity of B_Ψ, and dividing through by ηT gives the claimed result. ■

It is straightforward to transform Theorem 5 into an in-expectation result for stochastic subgradients:

Corollary 6 *Take* $f_t : \mathcal{W} \to \mathbb{R}$ *to be a sequence of convex functions, and* $\mathcal{F}$ *a filtration. Suppose that we perform* T *iterations of stochastic mirror descent starting from* $w^{(1)} \in \mathcal{W}$*, using the definitions of Theorem 5:*

$$\begin{aligned}\tilde{w}^{(t+1)} &= \nabla\Psi^*\left(\nabla\Psi\left(w^{(t)}\right) - \eta\check{\Delta}^{(t)}\right), \\ w^{(t+1)} &= \underset{w\in\mathcal{W}}{\operatorname{argmin}} B_\Psi\left(w \,\middle|\, \tilde{w}^{(t+1)}\right),\end{aligned}$$

where $\check{\Delta}^{(t)}$ is a stochastic subgradient of f_t, i.e. $\mathbb{E}[\check{\Delta}^{(t)} \mid \mathcal{F}_{t-1}] \in \partial f_t(w^{(t)})$, and $\check{\Delta}^{(t)}$ is $\mathcal{F}_t$-measurable. Then:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[f_t\left(w^{(t)}\right) - f_t\left(w^*\right)\right] \le \frac{B_\Psi\left(w^* \mid w^{(1)}\right)}{\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\check{\Delta}^{(t)}\right\|_*^2\right],$$

where $w^* \in \mathcal{W}$ is an arbitrary reference vector.

Proof Define $\tilde{f}_t\left(w\right) = \left\langle\check{\Delta}^{(t)}, w\right\rangle$, and observe that applying the non-stochastic MD algorithm of Theorem 5 to the sequence of functions $\tilde{f}_t$ results in the same sequence of iterates $w^{(t)}$ as does applying the above stochastic MD update to the sequence of functions f_t. Hence:

$$\frac{1}{T}\sum_{t=1}^{T}\left(\tilde{f}_t\left(w^{(t)}\right) - \tilde{f}_t\left(w^*\right)\right) \le \frac{B_\Psi\left(w^* \mid w^{(1)}\right)}{\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T}\left\|\check{\Delta}^{(t)}\right\|_*^2. \tag{5}$$

By convexity, $f_t(w^{(t)}) - f_t(w^*) \le \left\langle\mathbb{E}[\check{\Delta}^{(t)} \mid \mathcal{F}_{t-1}], w^{(t)} - w^*\right\rangle$, while $\tilde{f}_t(w^{(t)}) - \tilde{f}_t(w^*) = \left\langle\check{\Delta}^{(t)}, w^{(t)} - w^*\right\rangle$ by definition. Taking expectations of both sides of Equation 5 and plugging in these inequalities yields the claimed result. ■

We next prove a high-probability analogue of the Corollary 6, based on a martingale bound of Dzhaparidze and van Zanten (2001):

Corollary 7 *In addition to the assumptions of Corollary 6, suppose that, with probability $1 - \delta_\sigma$, σ satisfies the following uniformly for all t $\in \{1, \dots, T\}$:*

$$\left\|\mathbb{E}\left[\check{\Delta}^{(t)} \;\middle|\; \mathcal{F}_{t-1}\right] - \check{\Delta}^{(t)}\right\|_* \le \sigma.$$

Then, with probability $1 - \delta_\sigma - \delta$, the above σ bound will hold, and:

$$\frac{1}{T}\sum_{t=1}^{T}\left(f_t\left(w^{(t)}\right) - f_t\left(w^*\right)\right) \le \frac{B_\Psi\left(w^* \mid w^{(1)}\right)}{\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T}\left\|\check{\Delta}^{(t)}\right\|_*^2 + \frac{\sqrt{2}R_*\sigma\sqrt{\ln\frac{1}{\delta}}}{\sqrt{T}} + \frac{2R_*\sigma\ln\frac{1}{\delta}}{3T},$$

where $w^* \in \mathcal{W}$ is an arbitrary reference vector and $R_* \ge \sup_{w\in\mathcal{W}}\left\|w - w^*\right\|$ bounds the radius of $\mathcal{W}$ centered on w^*.

Proof Define $\tilde{f}_t\left(w\right) = \left\langle\check{\Delta}^{(t)}, w\right\rangle$ as in the proof of Corollary 6, and observe that Equation 5 continues to apply. Define a sequence of random variables $M_0 = 0$, $M_t = M_{t-1} + \left\langle\mathbb{E}[\check{\Delta}^{(t)} \mid \mathcal{F}_{t-1}] - \check{\Delta}^{(t)}, w^{(t)} - w^*\right\rangle$, and notice that M forms a martingale w.r.t. the filtration $\mathcal{F}$. From this definition, Hölder's inequality gives that:

$$\left|M_t - M_{t-1}\right| \le \left\|\mathbb{E}\left[\check{\Delta}^{(t)} \;\middle|\; \mathcal{F}_{t-1}\right] - \check{\Delta}^{(t)}\right\|_* \left\|w^{(t)} - w^*\right\| \le R_*\sigma.$$

the above holding with probability $1 - \delta_\sigma$. Plugging $a = R_*\sigma$ and $L = TR_*^2\sigma^2$ into the Bernstein-type martingale inequality of Dzhaparidze and van Zanten (2001, Theorem 3.3) gives:

$$\Pr\left\{\frac{1}{T}M_T \ge \epsilon\right\} \le \delta_\sigma + \exp\left(-\frac{3T\epsilon^2}{6R_*^2\sigma^2 + 2R_*\sigma\epsilon}\right).$$

Solving for ϵ using the quadratic formula and upper-bounding gives that, with probability $1-\delta_\sigma-\delta$:

$$\frac{1}{T}\sum_{t=1}^{T}\left\langle \mathbb{E}\left[\check{\Delta}^{(t)} \;\middle|\; \mathcal{F}_{t-1}\right] - \check{\Delta}^{(t)}, w^{(t)} - w^{*}\right\rangle \leq \frac{\sqrt{2}R_{*}\sigma\sqrt{\ln\frac{1}{\delta}}}{\sqrt{T}} + \frac{2R_{*}\sigma\ln\frac{1}{\delta}}{3T}.$$

As in the proof of Corollary 6, $f_t(w^{(t)}) - f_t(w^{*}) \leq \left\langle \mathbb{E}[\check{\Delta}^{(t)} \mid \mathcal{F}_{t-1}], w^{(t)} - w^{*}\right\rangle$, while $\tilde{f}_t(w^{(t)}) - \tilde{f}_t(w^{*}) = \left\langle \check{\Delta}^{(t)}, w^{(t)} - w^{*}\right\rangle$ by definition, which combined with Equation 5 yields the claimed result. ■

Algorithm 2 (LightTouch) jointly optimizes over two sets of parameters, for which the objective is convex in the first and linear (hence concave) in the second. The convergence rate will be determined from a saddle-point bound, which we derive from Corollary 7 by following Nemirovski et al. (2009); Rakhlin and Sridharan (2013), and simply applying it twice:

Corollary 8 *Let $\|\cdot\|_w$ and $\|\cdot\|_\alpha$ be norms with duals $\|\cdot\|_{w*}$ and $\|\cdot\|_{\alpha*}$. Suppose that Ψ_w and Ψ_α are 1-strongly convex w.r.t. $\|\cdot\|_w$ and $\|\cdot\|_\alpha$, have convex conjugates Ψ_w^{*} and Ψ_α^{*}, and associated Bregman divergences B_{Ψ_w} and B_{Ψ_α}, respectively.*

Take $f : \mathcal{W} \times \mathcal{A} \to \mathbb{R}$ to be convex in its first parameter and concave in its second, let $\mathcal{F}$ be a filtration, and suppose that we perform T iterations of MD:

$$\begin{aligned}
\tilde{w}^{(t+1)} &= \nabla\Psi_w^{*}\left(\nabla\Psi_w\left(w^{(t)}\right) - \eta\check{\Delta}_w^{(t)}\right), & \tilde{\alpha}^{(t+1)} &= \nabla\Psi_\alpha^{*}\left(\nabla\Psi_\alpha\left(\alpha^{(t)}\right) + \eta\hat{\Delta}_\alpha^{(t)}\right), \\
w^{(t+1)} &= \underset{w\in\mathcal{W}}{\mathrm{argmin}} B_{\Psi_w}\left(w \;\middle|\; \tilde{w}^{(t+1)}\right), & \alpha^{(t+1)} &= \underset{\alpha\in\mathcal{A}}{\mathrm{argmin}} B_{\Psi_\alpha}\left(\alpha \;\middle|\; \tilde{\alpha}^{(t+1)}\right),
\end{aligned}$$

where $\check{\Delta}_w^{(t)}$ is a stochastic subgradient of $f(w^{(t)}, \alpha^{(t)})$ w.r.t. its first parameter, and $\hat{\Delta}_\alpha^{(t)}$ a stochastic supergradient w.r.t. its second, with both $\check{\Delta}_w^{(t)}$ and $\hat{\Delta}_\alpha^{(t)}$ being $\mathcal{F}_t$-measurable. We assume that, with probabilities $1-\delta_{\sigma w}$ and $1-\delta_{\sigma\alpha}$ (respectively), σ_w^2 and σ_α^2 satisfy the following uniformly for all $t \in \{1,\dots,T\}$:

$$\left\|\mathbb{E}\left[\check{\Delta}_w^{(t)} \;\middle|\; \mathcal{F}_{t-1}\right] - \check{\Delta}_w^{(t)}\right\|_{w*} \leq \sigma_w \qquad \textit{and} \qquad \left\|\mathbb{E}\left[\hat{\Delta}_\alpha^{(t)} \;\middle|\; \mathcal{F}_{t-1}\right] - \check{\Delta}_w^{(t)}\right\|_{\alpha*} \leq \sigma_\alpha.$$

Under these conditions, with probability $1-\delta_{\sigma w}-\delta_{\sigma\alpha}-2\delta$, the above σ_w and σ_α bounds will hold, and:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\left(f\left(w^{(t)}, \alpha^{*}\right) - f\left(w^{*}, \alpha^{(t)}\right)\right) \\
&\quad \leq \frac{B_{\Psi_w}\left(w^{*} \mid w^{(1)}\right) + B_{\Psi_\alpha}\left(\alpha^{*} \mid \alpha^{(1)}\right)}{\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T}\left(\left\|\check{\Delta}_w^{(t)}\right\|_{w*}^{2} + \left\|\hat{\Delta}_\alpha^{(t)}\right\|_{\alpha*}^{2}\right) \\
&\quad + \frac{\sqrt{2}\left(R_{w*}\sigma_w + R_{\alpha*}\sigma_\alpha\right)\sqrt{\ln\frac{1}{\delta}}}{\sqrt{T}} + \frac{2\left(R_{w*}\sigma_w + R_{\alpha*}\sigma_\alpha\right)\ln\frac{1}{\delta}}{3T},
\end{aligned}$$

where $w^{*} \in \mathcal{W}$ and $\alpha^{*} \in \mathcal{A}$ are arbitrary reference vectors, and $R_{w*} \geq \|w - w^{*}\|_w$ and $R_{\alpha*} \geq \|\alpha - \alpha^{*}\|_\alpha$ bound the radii of $\mathcal{W}$ and $\mathcal{A}$ centered on w^{*} and α^{*}, respectively.

Proof This is a convex-concave saddle-point problem, which we will optimize by playing two convex optimization algorithms against each other, as in Nemirovski et al. (2009); Rakhlin and Sridharan (2013). By Corollary 7, with probability $1 - \delta_{\sigma w} - \delta$ and $1 - \delta_{\sigma\alpha} - \delta$, respectively:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\left(f\left(w^{(t)}, \alpha^{(t)}\right) - f\left(w^{*}, \alpha^{(t)}\right)\right) \\
&\quad\leq \frac{B_{\Psi_w}\left(w^{*} \mid w^{(1)}\right)}{\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T}\left\|\check{\Delta}_w^{(t)}\right\|_{w*}^{2} + \frac{\sqrt{2}R_{w*}\sigma_w\sqrt{\ln\frac{1}{\delta}}}{\sqrt{T}} + \frac{2R_{w*}\sigma_w\ln\frac{1}{\delta}}{3T}, \\
&\frac{1}{T}\sum_{t=1}^{T}\left(f\left(w^{(t)}, \alpha^{*}\right) - f\left(w^{(t)}, \alpha^{(t)}\right)\right) \\
&\quad\leq \frac{B_{\Psi_\alpha}\left(\alpha^{*} \mid \alpha^{(1)}\right)}{\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T}\left\|\hat{\Delta}_\alpha^{(t)}\right\|_{\alpha*}^{2} + \frac{\sqrt{2}R_{\alpha*}\sigma_\alpha\sqrt{\ln\frac{1}{\delta}}}{\sqrt{T}} + \frac{2R_{\alpha*}\sigma_\alpha\ln\frac{1}{\delta}}{3T}.
\end{aligned}$$

Adding these two inequalities gives the claimed result. ■

Appendix B. SGD for Strongly-Convex Functions

For λ-strongly convex objective functions, we can achieve a faster convergence rate for SGD by using the step sizes $\eta_t = 1/\lambda t$. Our eventual algorithm (Algorithm 3) for strongly-convex heavily-constrained optimization will proceed in two phases, with the second phase "picking up" where the first phase "left off", for which reason we present a convergence rate, based on Shalev-Shwartz et al. (2011, Lemma 2), that effectively starts at iteration T_0 by using the step sizes $\eta_t = 1/\lambda(T_0 + t)$:

Theorem 9 *Take $f_t : \mathcal{W} \to \mathbb{R}$ to be a sequence of λ-strongly convex functions on which we perform T iterations of stochastic gradient descent starting from $w^{(1)} \in \mathcal{W}$:*

$$w^{(t+1)} = \Pi_w\left(w^{(t)} - \eta_t \check{\nabla} f_t\left(w^{(t)}\right)\right),$$

where $\check{\nabla} f_t\left(w^{(t)}\right) \in \partial f_t\left(w^{(t)}\right)$ is a subgradient of f_t at $w^{(t)}$, and $\left\|\check{\nabla} f_t\left(w^{(t)}\right)\right\|_2 \leq G$ for all t. If we choose $\eta_t = \frac{1}{\lambda(T_0+t)}$ for some $T_0 \in \mathbb{N}$, then:

$$\frac{1}{T}\sum_{t=1}^{T}\left(f_t\left(w^{(t)}\right) - f_t\left(w^{*}\right)\right) \leq \frac{G^2\left(1 + \ln T\right)}{2\lambda T} + \frac{\lambda T_0}{2T}\left\|w^{(1)} - w^{*}\right\|_2^2,$$

where $w^{*} \in \mathcal{W}$ is an arbitrary reference vector and $G \geq \left\|\check{\nabla} f_t\left(w^{(t)}\right)\right\|_2$ bounds the subgradient norms for all t.

Proof This is nothing but a small tweak to Shalev-Shwartz et al. (2011, Lemma 2). Starting from Equations 10 and 11 of that proof:

$$\begin{aligned}
&\sum_{t=1}^{T}\left(f_t\left(w^{(t)}\right)-f_t\left(w^*\right)\right)\\
&\quad\leq\frac{G^2}{2}\sum_{t=1}^{T}\eta_t+\sum_{t=1}^{T}\left(\frac{1}{2\eta_t}-\frac{\lambda}{2}\right)\left\|w^{(t)}-w^*\right\|_2^2-\sum_{t=1}^{T}\frac{1}{2\eta_t}\left\|w^{(t+1)}-w^*\right\|_2^2.
\end{aligned}$$

Taking $\eta_t=\frac{1}{\lambda(T_0+t)}$:

$$\begin{aligned}
&\sum_{t=1}^{T}\left(f_t\left(w^{(t)}\right)-f_t\left(w^*\right)\right)\\
&\quad\leq\frac{G^2}{2\lambda}\left(\frac{1}{T_0+1}+\int_{t=T_0+1}^{T_0+T}\frac{dt}{t}\right)+\frac{\lambda T_0}{2}\left\|w^{(1)}-w^*\right\|_2^2-\frac{\lambda\left(T_0+T\right)}{2}\left\|w^{(T+1)}-w^*\right\|_2^2.
\end{aligned}$$

Dividing through by T, simplifying and bounding yields the claimed result. ■

As we did Appendix A, we convert this into a result for stochastic subgradients:

Corollary 10 *Take $f_t:\mathcal{W}\rightarrow\mathbb{R}$ to be a sequence of λ-strongly convex functions, and $\mathcal{F}$ a filtration. Suppose that we perform T iterations of stochastic gradient descent starting from $w^{(1)}\in\mathcal{W}$:*

$$w^{(t+1)}=\Pi_w\left(w^{(t)}-\eta_t\check{\Delta}^{(t)}\right),$$

where $\check{\Delta}^{(t)}$ is a stochastic subgradient of f_t, i.e. $\mathbb{E}[\check{\Delta}^{(t)}\mid\mathcal{F}_{t-1}]\in\partial f_t(w^{(t)})$, and $\check{\Delta}^{(t)}$ is $\mathcal{F}_t$-measurable. If we choose $\eta_t=\frac{1}{\lambda(T_0+t)}$ for some $T_0\in\mathbb{N}$, then:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[f_t\left(w^{(t)}\right)-f_t\left(w^*\right)\right]\leq\frac{G^2\left(1+\ln T\right)}{2\lambda T}+\frac{\lambda T_0}{2T}\left\|w^{(1)}-w^*\right\|_2^2,$$

where $w^*\in\mathcal{W}$ is an arbitrary reference vector and $G\geq\left\|\check{\Delta}^{(t)}\right\|_2$ bounds the stochastic subgradient norms for all t.

Proof Same proof technique as Corollary 6, but based on Theorem 9 rather than Theorem 5. ■

We now use this result to prove an in-expectation saddle point bound:

Corollary 11 *Let $\|\cdot\|_\alpha$ and $\|\cdot\|_{\alpha*}$ be a norm and its dual. Suppose that Ψ_α is 1-strongly convex w.r.t. $\|\cdot\|_\alpha$, and has convex conjugate Ψ_α^* and associated Bregman divergence B_{Ψ_α}.*

Take $f : \mathcal{W} \times \mathcal{A} \to \mathbb{R}$ to be λ-strongly convex in its first parameter and concave in its second, let $\mathcal{F}$ be a filtration, and suppose that we perform T iterations of SGD on w and MD on α:

$$w^{(t+1)} = \Pi_w \left(w^{(t)} - \frac{1}{\lambda \left(T_0 + t\right)} \check{\Delta}_w^{(t)} \right), \qquad \begin{aligned} \tilde{\alpha}^{(t+1)} &= \nabla \Psi_\alpha^* \left(\nabla \Psi_\alpha \left(\alpha^{(t)} \right) + \eta \hat{\Delta}_\alpha^{(t)} \right), \\ \alpha^{(t+1)} &= \operatorname*{argmin}_{\alpha \in \mathcal{A}} B_{\Psi_\alpha} \left(\alpha \;\middle|\; \tilde{\alpha}^{(t+1)} \right), \end{aligned}$$

where $\check{\Delta}_w^{(t)}$ is a stochastic subgradient of $f(w^{(t)}, \alpha^{(t)})$ w.r.t. its first parameter, and $\hat{\Delta}_\alpha^{(t)}$ a stochastic supergradient w.r.t. its second, with both $\check{\Delta}_w^{(t)}$ and $\hat{\Delta}_\alpha^{(t)}$ being $\mathcal{F}_t$-measurable. Then:

$$\begin{aligned} &\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \left[f\left(w^{(t)}, \alpha^* \right) - f\left(w^*, \alpha^{(t)} \right) \right] \\ &\quad \leq \frac{G_w^2 \left(1 + \ln T\right)}{2\lambda T} + \frac{\lambda T_0}{2T} \left\| w^{(1)} - w^* \right\|_2^2 + \frac{B_{\Psi_\alpha} \left(\alpha^* \mid \alpha^{(1)} \right)}{\eta T} + \frac{\eta}{2T} \sum_{t=1}^{T} \mathbb{E} \left[\left\| \hat{\Delta}_\alpha^{(t)} \right\|_{\alpha *}^2 \right], \end{aligned}$$

where $w^* \in \mathcal{W}$ and $\alpha^* \in \mathcal{A}$ are arbitrary reference vectors, and $G_w \geq \left\| \check{\Delta}_w^{(t)} \right\|_2$ bounds the stochastic subgradient norms w.r.t. w for all t.

Proof As we did in the proof of Corollary 8, we will play two convex optimization algorithms against each other. By Corollaries 10 and 6:

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \left[f\left(w^{(t)}, \alpha^{(t)} \right) - f\left(w^*, \alpha^{(t)} \right) \right] &\leq \frac{G_w^2 \left(1 + \ln T\right)}{2\lambda T} + \frac{\lambda T_0}{2T} \left\| w^{(1)} - w^* \right\|_2^2, \\ \frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \left[f\left(w^{(t)}, \alpha^* \right) - f\left(w^{(t)}, \alpha^{(t)} \right) \right] &\leq \frac{B_{\Psi_\alpha} \left(\alpha^* \mid \alpha^{(1)} \right)}{\eta T} + \frac{\eta}{2T} \sum_{t=1}^{T} \mathbb{E} \left[\left\| \hat{\Delta}_\alpha^{(t)} \right\|_{\alpha *}^2 \right], \end{aligned}$$

Adding these two inequalities gives the claimed result. ■

Appendix C. Analyses of FullTouch and LightTouch

We begin by proving that, if γ is sufficiently large, then optimizing the relaxed objective, and projecting the resulting solution, will bring us close to the optimum of the constrained objective.

Lemma 1 *In the setting of Section 2, suppose that f is L_f-Lipschitz, i.e. $|f(w) - f(w')| \leq L_f \|w - w'\|_2$ for all $w, w' \in \mathcal{W}$, and that there is a constant $\rho > 0$ such that if $g(w) = 0$ then $\left\| \check{\nabla} \right\|_2 \geq \rho$ for all $\check{\nabla} \in \partial g(w)$, where $\partial g(w)$ is the subdifferential of $g(w)$.*

For a parameter $\gamma > 0$, define:

$$h\left(w\right) = f\left(w\right) + \gamma \max\left\{0, g\left(w\right)\right\}.$$

If $\gamma > L_f/\rho$, then for any infeasible w (i.e. for which $g(w) > 0$):

$$h\left(w\right) > h\left(\Pi_g\left(w\right)\right) = f\left(\Pi_g\left(w\right)\right) \qquad \text{and} \qquad \left\|w - \Pi_g\left(w\right)\right\|_2 \le \frac{h\left(w\right) - h\left(\Pi_g\left(w\right)\right)}{\gamma\rho - L_f},$$

where $\Pi_g\left(w\right)$ is the projection of w onto the set $\{w \in \mathcal{W} : g(w) \le 0\}$ w.r.t. the Euclidean norm.

Proof Let $w \in \mathcal{W}$ be an arbitrary infeasible point. Because f is L_f-Lipschitz:

$$f\left(w\right) \ge f\left(\Pi_g\left(w\right)\right) - L_f \left\|w - \Pi_g\left(w\right)\right\|_2. \tag{6}$$

Since $\Pi_g(w)$ is the projection of w onto the constraints w.r.t. the Euclidean norm, we must have by the first order optimality conditions that there exists a $\nu \ge 0$ such that:

$$0 \in \partial \left\|w - \Pi_g\left(w\right)\right\|_2^2 + \nu\partial g\left(\Pi_g\left(w\right)\right).$$

This implies that $w - \Pi_g(w)$ is a scalar multiple of some $\check{\nabla} \in \partial g(\Pi_g(w))$. Because g is convex and $\Pi_g\left(w\right)$ is on the boundary, $g(w) \ge g(\Pi_g(w)) + \left\langle\check{\nabla}, w - \Pi_g(w)\right\rangle = \left\langle\check{\nabla}, w - \Pi_g(w)\right\rangle$, so:

$$g(w) \ge \rho \left\|w - \Pi_g(w)\right\|_2. \tag{7}$$

Combining the definition of h with Equations 6 and 7 yields:

$$h\left(w\right) \ge f\left(\Pi_g\left(w\right)\right) + \left(\gamma\rho - L_f\right) \left\|w - \Pi_g(w)\right\|_2.$$

Both claims follow immediately if $\gamma\rho > L_f$. ■

C.1. Analysis of FullTouch

We'll now use Lemma 1 and Corollary 7 to bound the convergence rate of SGD on the function h of Lemma 1 (this is FullTouch). Like the algorithm itself, the convergence rate is little different from that found by Mahdavi et al. (2012) (aside from the bound on $\left\|\bar{w} - \Pi_g(\bar{w})\right\|_2$), and is included here only for completeness.

Lemma 12 *Suppose that the conditions of Lemma 1 apply, with $g(w) = \max_i(g_i(w))$. Define $D_w \ge \sup_{w,w' \in \mathcal{W}} \left\|w - w'\right\|_2$ as the diameter of $\mathcal{W}$, $G_f \ge \left\|\check{\Delta}^{(t)}\right\|_2$ and $G_g \ge \left\|\check{\nabla}\max(0, g_i(w))\right\|_2$ as uniform upper bounds on the (stochastic) gradient magnitudes of f and the g_is, respectively.*

If we optimize Equation 1 using Algorithm 1 (FullTouch) with the step size:

$$\eta = \frac{D_w}{\left(G_f + \gamma G_g\right)\sqrt{T}},$$

then with probability $1 - \delta$:

$$f\left(\Pi_g\left(\bar{w}\right)\right) - f\left(w^*\right) \le h\left(\bar{w}\right) - h\left(w^*\right) \le U_F, \qquad \text{and} \qquad \left\|\bar{w} - \Pi_g\left(\bar{w}\right)\right\|_2 \le \frac{U_F}{\gamma\rho - L_f},$$

where $w^* \in \{w \in \mathcal{W} : \forall i.g_i(w) \le 0\}$ is an arbitrary constraint-satisfying reference vector, and:

$$U_F \le \left(1 + 2\sqrt{2}\right) D_w \left(G_f + \gamma G_g\right)\sqrt{1 + \ln\frac{1}{\delta}}\sqrt{\frac{1}{T}} + \frac{8 D_w G_f \ln\frac{1}{\delta}}{3T}.$$

Proof We choose $\Psi(w) = \|w\|_2^2/2$, for which the mirror descent update rule is precisely SGD. Because Ψ_w is (half of) the squared Euclidean norm, it is trivially 1-strongly convex w.r.t. the Euclidean norm, so $\|\cdot\| = \|\cdot\|_* = \|\cdot\|_2$. Furthermore, $B_\Psi(w^* \mid w^{(1)}) \leq D_w^2/2$ and $R_* \leq D_w$.

We may upper bound the 2-norm of our stochastic gradients as $\left\|\check{\Delta}_w^{(t)}\right\|_2 \leq G_f + \gamma G_g$. Only the f-portion of the objective is stochastic, so the error of the $\check{\Delta}_w^{(t)}$s can be trivially upper bounded, with probability 1, with $\sigma = 2G_f$. Hence, by Corollary 7 (taking $\mathcal{F}_t$ to be e.g. the smallest σ-algebra making $\check{\Delta}^{(t)}, \dots, \check{\Delta}^{(t)}$ measurable), with probability $1-\delta$:

$$\frac{1}{T}\sum_{t=1}^{T}\left(h\left(w^{(t)}\right) - h\left(w^*\right)\right) \leq \frac{D_w^2}{2\eta T} + \frac{\eta\left(G_f + \gamma G_g\right)^2}{2} + \frac{2\sqrt{2}D_w G_f\sqrt{\ln\frac{1}{\delta}}}{\sqrt{T}} + \frac{8D_w G_f \ln\frac{1}{\delta}}{3T}.$$

Plugging in the definition of η, moving the average defining $\bar{w}$ inside h by Jensen's inequality, substituting $f(w^*) = h(w^*)$ because w^* satisfies the constraints, applying Lemma 1 and simplifying yields the claimed result. ■

In terms of the number of iterations required to achieve some desired level of suboptimality, this bound on U_F may be expressed as:

Theorem 13 *Suppose that the conditions of Lemmas 1 and 12 apply, and that η is as defined in Lemma 12.*

If we optimize Equation 1 using T_ϵ iterations of Algorithm 1 (FullTouch):

$$T_\epsilon = O\left(\frac{D_w^2\left(G_f + \gamma G_g\right)^2 \ln\frac{1}{\delta}}{\epsilon^2}\right),$$

then $U_F \leq \epsilon$ with probability $1-\delta$. where $w^* \in \{w \in \mathcal{W} : \forall i. g_i(w) \leq 0\}$ is an arbitrary constraint-satisfying reference vector.

Proof Based on the bound of Lemma 12, define:

$$\begin{aligned} x =& \sqrt{T}, \\ c =& \frac{8}{3}D_w G_f \ln\frac{1}{\delta}, \\ b =& \left(1 + 2\sqrt{2}\right) D_w \left(G_f + \gamma G_g\right)\sqrt{1 + \ln\frac{1}{\delta}}, \\ a =& -\epsilon, \end{aligned}$$

and consider the polynomial $0 = ax^2 + bx + c$. Roots of this polynomial are xs for which $U_F = \epsilon$, while for xs larger than any root we'll have that $U_F \leq \epsilon$. Hence, we can bound the T required to ensure ϵ-suboptimality by bounding the roots of this polynomial. By the Fujiwara bound (Wikipedia, 2015):

$$T_\epsilon \leq \max\left\{\frac{4\left(9 + 4\sqrt{2}\right)D_w^2\left(G_f + \gamma G_g\right)^2\left(1 + \ln\frac{1}{\delta}\right)}{\epsilon^2}, \frac{16 D_w G_f \ln\frac{1}{\delta}}{3\epsilon}\right\}, \tag{8}$$

Table 5: New notation in Appendix C.2.

Symbol	Description	Definition
$\Vert\cdot\Vert_w$, $\Vert\cdot\Vert_{w*}$	Norm on $\mathcal{W}$ and its dual	$\Vert\cdot\Vert_w = \Vert\cdot\Vert_{w*} = \Vert\cdot\Vert_2$
$\Vert\cdot\Vert_p$, $\Vert\cdot\Vert_{p*}$	Norm on Δ^m and its dual	$\Vert\cdot\Vert_p = \Vert\cdot\Vert_1$, $\Vert\cdot\Vert_{p*} = \Vert\cdot\Vert_\infty$
Ψ_w, Ψ_w^*	A d.g.f. on $\mathcal{W}$ and its convex conjugate	$\Psi_w(w) = \Vert w\Vert_2^2 / 2$
Ψ_p, Ψ_p^*	A d.g.f. on Δ^m and its convex conjugate	$\Psi_p(p) = \sum_{i=1}^m p_i \ln p_i$
R_{w*}	Bound on w^*-centered radius of $\mathcal{W}$	$R_{w*} = D_w \geq \Vert w - w^*\Vert_2$
R_{p*}	Bound on p^*-centered radius of Δ^m	$R_{p*} = 1 \geq \Vert p - p^*\Vert_1$
σ_w	Bound on $\check{\Delta}_w$ error	$\sigma_w = G_f + \gamma G_g$
σ_p	Bound on $\hat{\Delta}_p$ error	$\sigma_p \geq \left\Vert \mathbb{E}[\hat{\Delta}_p^{(t)} \mid \mathcal{F}_{t-1}] - \hat{\Delta}_p^{(t)} \right\Vert_\infty$
$1 - \delta_{\sigma w}$	Probability that σ_w bound holds	$1 - \delta_{\sigma w} = 1$
$1 - \delta_{\sigma p}$	Probability that σ_p bound holds	

giving the claimed result. ■

C.2. Analysis of LightTouch

Because we use the reduced-variance algorithm of Johnson and Zhang (2013), and therefore update the remembered gradient μ one random coordinate at a time, we must first bound the maximum number of iterations over which a coordinate can go un-updated:

Lemma 14 *Consider a process which maintains a sequence of vectors $s^{(t)} \in \mathbb{N}^m$ for $t \in \{1, \dots, T\}$, where $s^{(1)}$ is initialized to zero and $s^{(t+1)}$ is derived from $s^{(t)}$ by independently sampling $k = |S_t| \leq m$ random indices $S_t \subseteq \{1, \dots, m\}$ uniformly without replacement, and then setting $s_j^{(t+1)} = t$ for $j \in S_t$ and $s_j^{(t+1)} = s_j^{(t)}$ for $j \notin S_t$. Then, with probability $1 - \delta$:*

$$\max_{t,j} \left(t - s_j^{(t)} \right) \leq 1 + \frac{2m}{k} \ln \left(\frac{2mT}{\delta} \right).$$

Proof This is closely related to the "coupon collector's problem" (Wikipedia, 2014). We will begin by partitioning time into contiguous size-n chunks, with $1, \dots, n$ forming the first chunk, $n + 1, \dots, 2n$ the second, and so on.

Within each chunk the probability that any particular index was never sampled is $((m-k)/m)^n$, so by the union bound the probability that any one of the m indices was never sampled is bounded by $m((m-k)/m)^n$:

$$m \left(\frac{m-k}{m} \right)^n \leq \exp \left(\ln m + n \ln \left(\frac{m-k}{m} \right) \right) \leq \exp \left(\ln m - \frac{nk}{m} \right).$$

Define $n = \lceil (m/k) \ln(2mT/\delta) \rceil$, so:

$$m \left(\frac{m-k}{m} \right)^n \leq \exp \left(\ln m - \ln \left(\frac{2mT}{\delta} \right) \right) \leq \frac{\delta}{2T}.$$

This shows that for this choice of n, the probability of there existing an index which is never sampled in some particular batch is bounded by $\delta/2T$. By the union bound, the probability of *any* of $\lceil T/n \rceil$ batches containing an index which is never sampled is bounded by $(\delta/2T)\lceil T/n \rceil \le (\delta/2n) + (\delta/2T) \le \delta$.

If every index is sampled within every batch, then over the first $n\lceil T/n \rceil \ge T$ steps, the most steps which could elapse over which a particular index is not sampled is $2n - 2$ (if the index is sampled on the first step of one chunk, and the last step of the next chunk), which implies the claimed result. ■

We now combine this bound with Corollary 7 and make appropriate choices of the two d.g.f.s to yield a bound on the LightTouch convergence rate:

Lemma 15 *Suppose that the conditions of Lemma 1 apply, with $g(w) = \max_i(g_i(w))$. Define $D_w \ge \sup_{w,w' \in \mathcal{W}} \max\{1, \|w - w'\|_2\}$ as a bound on the diameter of $\mathcal{W}$ (notice that we also choose D_w to be at least 1), $G_f \ge \left\|\check{\Delta}^{(t)}\right\|_2$ and $G_g \ge \left\|\check{\nabla} \max(0, g_i(w))\right\|_2$ as uniform upper bounds on the (stochastic) gradient magnitudes of f and the g_is, respectively, for all $i \in \{1, \dots, m\}$. We also assume that all g_is are L_g-Lipschitz w.r.t. $\|\cdot\|_2$, i.e. $|g_i(w) - g_i(w')| \le L_g \|w - w'\|_2$ for all $w, w' \in \mathcal{W}$.*

Define:

$$k = \left\lceil \frac{m\left(1 + \ln m\right)^{3/4} \sqrt{1 + \ln \frac{1}{\delta}} \sqrt{1 + \ln T}}{T^{1/4}} \right\rceil .$$

If $k \le m$ and we optimize Equation 1 using Algorithm 2 (LightTouch), basing the stochastic gradients w.r.t. p on k constraints at each iteration, and using the step size:

$$\eta = \frac{\sqrt{1 + \ln m} D_w}{\left(G_f + \gamma G_g + \gamma L_g D_w\right) \sqrt{T}},$$

then it holds with probability $1 - \delta$ that:

$$f\left(\Pi_g\left(\bar{w}\right)\right) - f\left(w^*\right) \le h\left(\bar{w}\right) - h\left(w^*\right) \le U_L, \qquad \text{and} \qquad \left\|\bar{w} - \Pi_g\left(\bar{w}\right)\right\|_2 \le \frac{U_L}{\gamma\rho - L_f},$$

where $w^* \in \{w \in \mathcal{W} : \forall i. g_i(w) \le 0\}$ is an arbitrary constraint-satisfying reference vector, and:

$$U_L \le 67\sqrt{1 + \ln m} D_w \left(G_f + \gamma G_g + \gamma L_g D_w\right) \sqrt{1 + \ln \frac{1}{\delta}} \sqrt{\frac{1}{T}}.$$

If $k > m$, then we should fall-back to using FullTouch, in which case the result of Lemma 12 will apply.

Proof We choose $\Psi_w(w) = \|w\|_2^2 / 2$ and $\Psi_p(p) = \sum_{i=1}^m p_i \ln p_i$ to be the squared Euclidean norm divided by 2 and the negative Shannon entropy, respectively, which yields the updates of Algorithm 2. We assume that the $\check{\Delta}^{(t)}$s are random variables on some probability space (depending on the source of the stochastic gradients of f), and likewise the i_ts and j_ts on another, so $\mathcal{F}_t$ may be taken to be the product of the smallest σ-algebras which make $\check{\Delta}^{(1)}, \dots, \check{\Delta}^{(t)}$ and $i_1, j_1, \dots, i_t, j_t$ measurable, respectively, with conditional expectations being taken w.r.t. the product measure. Under the definitions of Corollary 8 (taking $\alpha = p$), with probability $1 - \delta_{\sigma w} - \delta_{\sigma p} - 2\delta'$:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T} \tilde{h}\left(w^{(t)}, p^*\right) - \frac{1}{T}\sum_{t=1}^{T} \tilde{h}\left(w^*, p^{(t)}\right) \\
&\leq \frac{B_{\Psi_w}\left(w^* \mid w^{(1)}\right) + B_{\Psi_p}\left(p^* \mid p^{(1)}\right)}{\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T}\left(\left\|\check{\Delta}_w^{(t)}\right\|_{w*}^2 + \left\|\hat{\Delta}_p^{(t)}\right\|_{p*}^2\right) \\
&\quad + \frac{\sqrt{2}\left(R_{w*}\sigma_w + R_{p*}\sigma_p\right)\sqrt{\ln\frac{1}{\delta'}}}{\sqrt{T}} + \frac{4\left(R_{w*}\sigma_w + R_{p*}\sigma_p\right)\ln\frac{1}{\delta'}}{3T}.
\end{aligned}$$

As in the proof of Lemma 12, Ψ_w is 1-strongly convex w.r.t. the Euclidean norm, so $\|\cdot\|_w = \|\cdot\|_{w*} = \|\cdot\|_2$, $B_{\Psi_w}(w^* \mid w^{(1)}) \leq D_w^2/2$ and $R_{w*} \leq D_w$. Because Ψ_p is the negative entropy, which is 1-strongly convex w.r.t. the 1-norm (this is Pinsker's inequality), $\|\cdot\|_p = \|\cdot\|_1$ and $\|\cdot\|_{p*} = \|\cdot\|_\infty$, implying that $R_{p*} = 1$. Since $p^{(1)}$ is initialized to the uniform distribution, $B_{\Psi_p}(p^* \mid p^{(1)}) = D_{KL}(p^* \mid p^{(1)}) \leq \ln m$.

The stochastic gradient definitions of Algorithm 2 give that $\left\|\check{\Delta}_w^{(t)}\right\|_{w*} \leq G_f + \gamma G_g$ and $\sigma_w \leq 2(G_f + \gamma G_g)$ with probability $1 = 1 - \delta_{\sigma w}$ by the triangle inequality, and $\tilde{h}(w^*, p^{(t)}) = f(w^*)$ because w^* satisfies the constraints. All of these facts together give that, with probability $1 - \delta_{\sigma p} - \delta'$:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T} \tilde{h}\left(w^{(t)}, p^*\right) - f\left(w^*\right) \\
&\leq \frac{D_w^2 + 2\ln m}{2\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T}\left(\left(G_f + \gamma G_g\right)^2 + \left\|\hat{\Delta}_p^{(t)}\right\|_\infty^2\right) \\
&\quad + \frac{\sqrt{2}\left(2D_w(G_f + \gamma G_g) + \sigma_p\right)\sqrt{\ln\frac{1}{\delta'}}}{\sqrt{T}} + \frac{4\left(2D_w\left(G_f + \gamma G_g\right) + \sigma_p\right)\ln\frac{1}{\delta'}}{3T}.
\end{aligned}$$

We now move the average defining $\bar{w}$ inside $\tilde{h}$ (which is convex in its first parameter) by Jensen's inequality, and use the fact that there exists a p^* such that $\tilde{h}(w, p^*) = h(w)$ to apply Lemma 1:

$$\begin{aligned}
U_L \leq{}& \frac{D_w^2 + 2\ln m}{2\eta T} + \frac{\eta}{2T}\sum_{t=1}^{T}\left(\left(G_f + \gamma G_g\right)^2 + \left\|\hat{\Delta}_p^{(t)}\right\|_\infty^2\right) \\
&+ \frac{\sqrt{2}\left(2D_w(G_f + \gamma G_g) + \sigma_p\right)\sqrt{\ln\frac{1}{\delta'}}}{\sqrt{T}} + \frac{4\left(2D_w\left(G_f + \gamma G_g\right) + \sigma_p\right)\ln\frac{1}{\delta'}}{3T}.
\end{aligned} \tag{9}$$

By the triangle inequality and the fact that $(a+b)^2 \leq 2a^2 + 2b^2$:

$$\begin{aligned}
\left\|\hat{\Delta}_p^{(t)}\right\|_\infty^2 &\leq 2\left\|\mathbb{E}\left[\hat{\Delta}_p^{(t)} \,\middle|\, \mathcal{F}_{t-1}\right]\right\|_\infty^2 + 2\left\|\mathbb{E}\left[\hat{\Delta}_p^{(t)} \,\middle|\, \mathcal{F}_{t-1}\right] - \hat{\Delta}_p^{(t)}\right\|_\infty^2 \\
&\leq 2\gamma^2 L_g^2 D_w^2 + 2\left\|\mathbb{E}\left[\hat{\Delta}_p^{(t)} \,\middle|\, \mathcal{F}_{t-1}\right] - \hat{\Delta}_p^{(t)}\right\|_\infty^2 \\
&\leq 2\gamma^2 L_g^2 D_w^2 + 2\sigma_p^2.
\end{aligned}$$

Substituting into Equation 9 and using the fact that $a + b \leq (\sqrt{a} + \sqrt{b})^2$:

$$U_L \leq \frac{D_w^2 + 2\ln m}{2\eta T} + \frac{\eta}{2}\left(G_f + \gamma G_g + \sqrt{2}\gamma L_g D_w\right)^2 + \eta\sigma_p^2 \\ + \frac{\sqrt{2}\left(2D_w(G_f + \gamma G_g) + \sigma_p\right)\sqrt{\ln\frac{1}{\delta'}}}{\sqrt{T}} + \frac{4\left(2D_w\left(G_f + \gamma G_g\right) + \sigma_p\right)\ln\frac{1}{\delta'}}{3T}. \tag{10}$$

We will now turn our attention to the problem of bounding σ_p. Notice that because we sample *i.i.d.* j_ts uniformly at every iteration, they form an instance of the process of Lemma 14 with $\mu_j^{(t)} = \max(0, g_j(w^{(s_j^{(t)})}))$, showing that with probability $1 - \delta_{\sigma p}$:

$$\max_{t,j}\left(t - s_j^{(t)}\right) \leq 1 + \frac{2m}{k}\ln\left(\frac{2mT}{\delta_{\sigma p}}\right). \tag{11}$$

By the definition of $\hat{\Delta}_p^{(t)}$ (Algorithm 2):

$$\begin{aligned}
&\left\|\mathbb{E}\left[\hat{\Delta}_p^{(t)} \,\middle|\, \mathcal{F}_{t-1}\right] - \hat{\Delta}_p^{(t)}\right\|_\infty^2 \\
=&\gamma^2\left\|\left(\sum_{j=1}^m e_j \max\left\{0, g_j\left(w^{(t)}\right)\right\} - \mu^{(t)}\right) - \frac{m}{k}\sum_{j\in S_t}\left(e_j\max\left\{0, g_j\left(w^{(t)}\right)\right\} - e_j\mu_j^{(t)}\right)\right\|_\infty^2 \\
\leq&\gamma^2\left(\frac{m-k}{k}\right)^2\max_j\left(\max\left\{0, g_j\left(w^{(t)}\right)\right\} - \mu_j^{(t)}\right)^2 \\
\leq&\gamma^2\left(\frac{m-k}{k}\right)^2 L_g^2\left\|w^{(t)} - w^{(s_j^{(t)})}\right\|_2^2 \\
\leq&\gamma^2\left(\frac{m-k}{k}\right)^2 L_g^2\eta^2\left(G_f + \gamma G_g\right)^2\left(t - s_j^{(t)}\right)^2 \\
\leq&\gamma^2\left(\frac{m-k}{k}\right)^2 L_g^2\eta^2\left(G_f + \gamma G_g\right)^2\left(1 + \frac{2m}{k}\ln\left(\frac{2mT}{\delta_{\sigma p}}\right)\right)^2 \\
\leq&6\gamma^2\left(\frac{m}{k}\right)^4 L_g^2\eta^2\left(G_f + \gamma G_g\right)^2\left(1 + \ln\left(\frac{mT}{\delta_{\sigma p}}\right)\right)^2
\end{aligned}$$

where in the second step we used the definition of the ∞-norm, in the third we used the Lipschitz continuity of the g_is (and hence of their positive parts), in the fourth we bounded the distance between two iterates with the number of iterations times a bound on the total step size, and in the fifth we used Equation 11. This shows that we may define:

$$\sigma_p = \sqrt{6}\gamma\left(\frac{m}{k}\right)^2 L_g\eta\left(G_f + \gamma G_g\right)\left(1 + \ln\left(\frac{mT}{\delta_{\sigma p}}\right)\right),$$

and it will satisfy the conditions of Corollary 8. Notice that, due to the η factor, σ_p will be *decreasing* in T. Substituting the definitions of η and σ_p into Equation 10, choosing $\delta_{\sigma p} = \delta' = \delta/3$ and using the assumption that $D_w \geq 1$ gives that with probability $1-\delta$:

$$
\begin{aligned}
U_L \leq & 2\left(1+\sqrt{2}\right)\sqrt{1+\ln 3}\sqrt{1+\ln m} D_w \left(G_f + \gamma G_g + \gamma L_g D_w\right)\sqrt{1+\ln\frac{1}{\delta}}\left(\frac{1}{\sqrt{T}}\right) \\
&+ \left(2\sqrt{3}+\frac{8}{3}\right)\left(1+\ln 3\right)^{3/2}\left(\frac{m}{k}\right)^2 \left(1+\ln m\right)^{3/2} D_w \left(G_f+\gamma G_g\right)\left(1+\ln\frac{1}{\delta}\right)^{3/2}\left(\frac{1+\ln T}{T}\right) \\
&+ 2\left(3+2\sqrt{\frac{2}{3}}\right)\left(1+\ln 3\right)^2\left(\frac{m}{k}\right)^4\left(1+\ln m\right)^{7/2} D_w\left(G_f+\gamma G_g\right)\left(1+\ln\frac{1}{\delta}\right)^2\left(\frac{\left(1+\ln T\right)^2}{T^{3/2}}\right).
\end{aligned}
$$

Rounding up the constant terms:

$$
\begin{aligned}
U_L \leq & 7\sqrt{1+\ln m} D_w \left(G_f + \gamma G_g + \gamma L_g D_w\right)\sqrt{1+\ln\frac{1}{\delta}}\left(\frac{1}{\sqrt{T}}\right) \\
&+ 19\left(\frac{m}{k}\right)^2 \left(1+\ln m\right)^{3/2} D_w \left(G_f+\gamma G_g\right)\left(1+\ln\frac{1}{\delta}\right)^{3/2}\left(\frac{1+\ln T}{T}\right) \\
&+ 41\left(\frac{m}{k}\right)^4\left(1+\ln m\right)^{7/2} D_w\left(G_f+\gamma G_g\right)\left(1+\ln\frac{1}{\delta}\right)^2\left(\frac{\left(1+\ln T\right)^2}{T^{3/2}}\right).
\end{aligned}
$$

Substituting the definition of k, simplifying and bounding yields the claimed result. ■

In terms of the number of iterations required to achieve some desired level of suboptimality, this bound on U_L and the bound of Lemma 12 on U_F may be combined to yield the following:

Theorem 3 *Suppose that the conditions of Lemmas 1 and 15 apply. Our result will be expressed in terms of a total iteration count T_ϵ satisfying:*

$$
T_\epsilon = O\left(\frac{\left(\ln m\right) D_w^2 \left(G_f+\gamma G_g+\gamma L_g D_w\right)^2 \ln\frac{1}{\delta}}{\epsilon^2}\right).
$$

Define k in terms of T_ϵ as in Lemma 15. If $k \leq m$, then we optimize Equation 1 using T_ϵ iterations of Algorithm 2 (LightTouch) with η as in Lemma 15. If $k > m$, then we use T_ϵ iterations of Algorithm 1 (FullTouch) with η as in Lemma 12. In either case, we perform T_ϵ iterations, requiring a total of C_ϵ "constraint checks" (evaluations or differentiations of a single g_i):

$$
\begin{aligned}
C_\epsilon = & \tilde{O}\left(\frac{\left(\ln m\right) D_w^2 \left(G_f+\gamma G_g+\gamma L_g D_w\right)^2 \ln\frac{1}{\delta}}{\epsilon^2}\right. \\
&\left. + \frac{m\left(\ln m\right)^{3/2} D_w^{3/2}\left(G_f+\gamma G_g+\gamma L_g D_w\right)^{3/2}\left(\ln\frac{1}{\delta}\right)^{5/4}}{\epsilon^{3/2}}\right).
\end{aligned}
$$

and with probability $1-\delta$:

$$
f\left(\Pi_g\left(\bar{w}\right)\right) - f\left(w^*\right) \leq h\left(\bar{w}\right) - h\left(w^*\right) \leq \epsilon \qquad \textit{and} \qquad \left\|\bar{w}-\Pi_g\left(\bar{w}\right)\right\|_2 \leq \frac{\epsilon}{\gamma\rho - L_f},
$$

where $w^* \in \{w \in \mathcal{W} : \forall i . g_i(w) \leq 0\}$ is an arbitrary constraint-satisfying reference vector.

Proof Regardless of the value of k, it follows from Lemmas 15 and 12 that:

$$U_L, U_F \leq 67\sqrt{1+\ln m} D_w \left(G_f + \gamma G_g + \gamma L_g D_w\right) \sqrt{1+\ln\frac{1}{\delta}} \sqrt{\frac{1}{T}} + \frac{8 D_w G_f \ln\frac{1}{\delta}}{3T}.$$

As in the proof of Theorem 13, we define:

$$\begin{aligned} x =& \sqrt{T}, \\ c =& \frac{8}{3} D_w G_f \ln\frac{1}{\delta}, \\ b =& 67\sqrt{1+\ln m} D_w \left(G_f + \gamma G_g + \gamma L_g D_w\right) \sqrt{1+\ln\frac{1}{\delta}} \sqrt{\frac{1}{T}}, \\ a =& -\epsilon, \end{aligned}$$

and consider the polynomial $0 = ax^2 + bx + c$. Any upper bound on all roots $x = \sqrt{T}$ of this polynomial will result in a lower-bound the values of T for which $U_L, U_F \leq \epsilon$ with probability $1-\delta$. By the Fujiwara bound (Wikipedia, 2015):

$$T_\epsilon = \max\left\{ \frac{(134)^2 \left(1+\ln m\right) D_w^2 \left(G_f + \gamma G_g + \gamma L_g D_w\right)^2 \left(1+\ln\frac{1}{\delta}\right)}{\epsilon^2}, \frac{16 D_w G_f \ln\frac{1}{\delta}}{3\epsilon} \right\},$$

giving the claimed bound on T_ϵ. For C_ϵ, we observe that we will perform no more than $k+1$ constraint checks at each iteration ($k+1$ by LightTouch if $k \leq m$, and $m+1$ by FullTouch if $k > m$), and substitute the above bound on T_ϵ into the definition of k, yielding:

$$\begin{aligned} \left(k+1\right) T_\epsilon \leq& 2T_\epsilon + m\left(1+\ln m\right)^{3/4} \sqrt{1+\ln\frac{1}{\delta}} T_\epsilon^{3/4} \sqrt{1+\ln T_\epsilon} \\ \leq& \max\left\{ \frac{2\left(134\right)^2 \left(1+\ln m\right) D_w^2 \left(G_f + \gamma G_g + \gamma L_g D_w\right)^2 \left(1+\ln\frac{1}{\delta}\right)}{\epsilon^2}, \frac{32 D_w G_f \ln\frac{1}{\delta}}{3\epsilon} \right\} \\ &+ \max\left\{ \frac{\left(134\right)^{3/2} m \left(1+\ln m\right)^{3/2} D_w^{3/2} \left(G_f + \gamma G_g + \gamma L_g D_w\right)^{3/2} \left(1+\ln\frac{1}{\delta}\right)^{5/4}}{\epsilon^{3/2}}, \right. \\ &\left. \left(\frac{16}{3}\right)^{3/4} \frac{m\left(1+\ln m\right)^{3/4} D_w^{3/4} G_f^{3/4} \left(1+\ln\frac{1}{\delta}\right)^{5/4}}{\epsilon^{3/4}} \right\} \sqrt{1+\ln T_\epsilon}. \end{aligned}$$

giving the claimed result (notice the $\sqrt{1+\ln T_\epsilon}$ factor on the RHS, for which reason we have a $\tilde{O}$ bound on C_ϵ, instead of O). ■

Appendix D. Analysis of MidTouch

We now move on to the analysis of our LightTouch variant for λ-strongly convex objectives, Algorithm 3 (MidTouch). While we were able to prove a high-probability bound for LightTouch, we were unable to do so for MidTouch, because the extra terms resulting from the use of a Bernstein-type martingale inequality were too large (since the other terms shrank as a result of the strong convexity assumption). Instead, we give an in-expectation result, and leave the proof of a corresponding high-probability bound to future work.

Our first result is an analogue of Lemmas 12 and 15, and bounds the suboptimality achieved by MidTouch as a function of the iteration counts T_1 and T_2 of the two phases:

Lemma 16 *Suppose that the conditions of Lemma 1 apply, with $g(w) = \max_i(g_i(w))$. Define $G_f \geq \left\| \check{\Delta}^{(t)} \right\|_2$ and $G_g \geq \left\| \check{\nabla} \max(0, g_i(w)) \right\|_2$ as uniform upper bounds on the (stochastic) gradient magnitudes of f and the g_is, respectively, for all $i \in \{1, \dots, m\}$. We also assume that f is λ-strongly convex, and that all g_is are L_g-Lipschitz w.r.t. $\|\cdot\|_2$, i.e. $|g_i(w) - g_i(w')| \leq L_g \|w - w'\|_2$ for all $w, w' \in \mathcal{W}$.*

If we optimize Equation 1 using Algorithm 3 (MidTouch) with the p-update step size $\eta = \lambda / 2\gamma^2 L_g^2$, then:

$$\begin{aligned}
\mathbb{E}\left[\|\Pi_g(\bar{w}) - w^*\|_2^2 \right] &\leq \mathbb{E}\left[\|\bar{w} - w^*\|_2^2 \right] \\
&\leq \frac{2\left(G_f + \gamma G_g\right)^2 (2 + \ln T_1 + \ln T_2) + 8\gamma^2 L_g^2 \ln m}{\lambda^2 T_2} + \frac{3m^4 (1 + \ln m)^2 \left(G_f + \gamma G_g\right)^2}{\lambda^2 T_1^2},
\end{aligned}$$

where $w^* = \operatorname{argmin}_{\{w \in \mathcal{W} : \forall i . g_i(w) \leq 0\}} f(w)$ is the optimal *constraint-satisfying reference vector.*

Proof As in the proof of Lemma 12, the first phase of Algorithm 3 is nothing but (strongly convex) SGD on the overall objective function h, so by Corollary 10:

$$\frac{1}{T_1} \sum_{t=1}^{T_1} \mathbb{E}\left[h\left(w^{(t)}\right) - h\left(w^*\right) \right] \leq \frac{G_w^2 \left(1 + \ln T_1\right)}{2\lambda T_1},$$

so by Jensen's inequality:

$$\mathbb{E}\left[h\left(w^{(T_1+1)}\right) - h\left(w^*\right) \right] \leq \frac{G_w^2 \left(1 + \ln T_1\right)}{2\lambda T_1}. \tag{12}$$

For the second phase, as in the proof of Lemma 15, we choose $\Psi_p(p) = \sum_{i=1}^m p_i \ln p_i$ to be negative Shannon entropy, which yields the second-phase updates of Algorithm 3. By Corollary 11:

$$\begin{aligned}
&\frac{1}{T_2} \sum_{t=T_1+1}^{T_2} \mathbb{E}\left[\tilde{h}\left(w^{(t)}, p^*\right) - \tilde{h}\left(w^*, p^{(t)}\right) \right] \\
&\leq \frac{G_w^2 \left(1 + \ln T\right)}{2\lambda T_2} + \frac{\lambda T_1}{2T_2} \left\| w^{(T_1+1)} - w^* \right\|_2^2 + \frac{B_{\Psi_p}\left(p^* \mid p^{(T_1+1)}\right)}{\eta T_2} + \frac{\eta}{2T_2} \sum_{t=T_1+1}^{T_2} \mathbb{E}\left[\left\| \hat{\Delta}_p^{(t)} \right\|_{p*}^2 \right].
\end{aligned}$$

As before, $\|\cdot\|_p = \|\cdot\|_1$, $\|\cdot\|_{p*} = \|\cdot\|_\infty$, and $B_{\Psi_p}(p^* \mid p^{(T_1+1)}) = D_{KL}(p^* \mid p^{(T_1+1)}) \leq \ln m$. Hence:

$$\begin{aligned}&\frac{1}{T_2}\sum_{t=T_1+1}^{T_2} \mathbb{E}\left[\tilde{h}\left(w^{(t)}, p^*\right) - \tilde{h}\left(w^*, p^{(t)}\right)\right] \\ &\quad\leq \frac{G_w^2\left(1+\ln T_2\right)}{2\lambda T_2} + \frac{\lambda T_1}{2T_2}\left\|w^{(T_1+1)} - w^*\right\|_2^2 + \frac{\ln m}{\eta T_2} + \frac{\eta}{2T_2}\sum_{t=T_1+1}^{T_2}\mathbb{E}\left[\left\|\hat{\Delta}_p^{(t)}\right\|_\infty^2\right].\end{aligned}$$

Since h is λ-strongly convex and w^* is optimal, $\left\|w^{(T_1+1)} - w^*\right\|_2^2 \leq \frac{2}{\lambda}(h(w^{(T_1+1)}) - h(w^*))$. By Equation 12:

$$\begin{aligned}&\frac{1}{T_2}\sum_{t=T_1+1}^{T_2} \mathbb{E}\left[\tilde{h}\left(w^{(t)}, p^*\right) - \tilde{h}\left(w^*, p^{(t)}\right)\right] \\ &\quad\leq \frac{G_w^2\left(2+\ln T_1 + \ln T_2\right)}{2\lambda T_2} + \frac{\ln m}{\eta T_2} + \frac{\eta}{2T_2}\sum_{t=T_1+1}^{T_2}\mathbb{E}\left[\left\|\hat{\Delta}_p^{(t)}\right\|_\infty^2\right].\end{aligned}$$

Since the (uncentered) second moment is equal to the mean plus the variance, and using the fact that $\tilde{h}(w^*, p^{(t)}) = f(w^*)$ since all constraints are satisfied at w^*:

$$\begin{aligned}&\frac{1}{T_2}\sum_{t=T_1+1}^{T_2} \mathbb{E}\left[\tilde{h}\left(w^{(t)}, p^*\right)\right] - f\left(w^*\right) \\ &\quad\leq \frac{G_w^2\left(2+\ln T_1 + \ln T_2\right)}{2\lambda T_2} + \frac{\ln m}{\eta T_2} + \frac{\eta}{2T_2}\sum_{t=T_1+1}^{T_2}\left(\mathbb{E}\left[\left\|\hat{\Delta}_p^{(t)}\right\|_\infty\right]\right)^2 + \frac{\eta\sigma_p^2}{2},\end{aligned} \tag{13}$$

where σ_p^2 is the variance of $\left\|\hat{\Delta}_p^{(t)}\right\|_\infty$. Next observe that:

$$\begin{aligned}\left(\mathbb{E}\left[\left\|\hat{\Delta}_p^{(t)}\right\|_\infty\right]\right)^2 =& \left(\mathbb{E}\left[\max_{j\in\{1,\dots,m\}}\gamma\max\left\{0, g_j\left(w^{(t)}\right)\right\}\right]\right)^2 \\ \leq& \gamma^2 L_g^2 \mathbb{E}\left[\left\|w^{(t)} - w^*\right\|_2^2\right] \\ \leq& \frac{2\gamma^2 L_g^2}{\lambda}\mathbb{E}\left[\tilde{h}\left(w^{(t)}, p^*\right) - \tilde{h}\left(w^*, p^*\right)\right],\end{aligned}$$

the first step using the fact that the g_js are L_g-Lipschitz and Jensen's inequality. For the second step, we choose p^* such that w^*, p^* is a minimax optimal pair (recall that w^* is optimal by assumption), and use the λ-strong convexity of $\tilde{h}$. Substituting into Equation 13 and using the fact that $\tilde{h}(w^*, p^*) = f(w^*)$:

$$\left(1 - \frac{\eta\gamma^2 L_g^2}{\lambda}\right)\left(\frac{1}{T_2}\sum_{t=T_1+1}^{T_2}\mathbb{E}\left[\tilde{h}\left(w^{(t)}, p^*\right)\right] - f\left(w^*\right)\right) \leq \frac{G_w^2\left(2+\ln T_1+\ln T_2\right)}{2\lambda T_2} + \frac{\ln m}{\eta T_2} + \frac{\eta\sigma_p^2}{2}.$$

Substituting $\eta = \lambda / 2\gamma^2 L_g^2$ and using Jensen's inequality:

$$\mathbb{E}\left[\tilde{h}\left(\bar{w}, p^*\right)\right] - f\left(w^*\right) \leq \frac{G_w^2\left(2 + \ln T_1 + \ln T_2\right)}{\lambda T_2} + \frac{4\gamma^2 L_g^2 \ln m}{\lambda T_2} + \frac{\lambda \sigma_p^2}{2\gamma^2 L_g^2}. \tag{14}$$

We now follow the proof of Lemma 15 and bound σ_p^2. By the definition of $\hat{\Delta}_p^{(t)}$ (Algorithm 3):

$$\begin{aligned}
\sigma_p^2 =& \mathbb{E}\left[\left\| \mathbb{E}\left[\hat{\Delta}_p^{(t)} \;\middle|\; \mathcal{F}_{t-1}\right] - \hat{\Delta}_p^{(t)} \right\|_\infty^2\right] \\
=& \gamma^2 \mathbb{E}\left[\left\| \left(\sum_{j=1}^m e_j \max\left\{0, g_j\left(w^{(t)}\right)\right\} - \mu^{(t)}\right) - m\left(e_{j_t} \max\left\{0, g_{j_t}\left(w^{(t)}\right)\right\} - e_{j_t}\mu_{j_t}^{(t)}\right) \right\|_\infty^2\right] \\
\leq& \gamma^2 \left(m-1\right)^2 \mathbb{E}\left[\max_j \left(\max\left\{0, g_j\left(w^{(t)}\right)\right\} - \mu_j^{(t)}\right)^2\right].
\end{aligned}$$

The indices j are sampled uniformly, so the maximum time $\max_j(t - s_j^{(t)})$ since we last sampled the same index is an instance of the coupon collector's problem Wikipedia (2014). Because the g_js are L_g-Lipschitz:

$$\begin{aligned}
\sigma_p^2 \leq& \gamma^2 \left(m-1\right)^2 L_g^2 \mathbb{E}\left[\max_j \left\| w^{(t)} - w^{\left(s_j^{(t)}\right)} \right\|_2^2\right] \\
\leq& \frac{\gamma^2 \left(m-1\right)^2 L_g^2 G_w^2}{\lambda^2 T_1^2} \mathbb{E}\left[\max_j \left(t - s_j^{(t)}\right)^2\right] \\
\leq& \frac{\gamma^2 m^4 \left(1 + \left(\ln m\right)^2 + \pi^2/6\right) L_g^2 G_w^2}{\lambda^2 T_1^2} \\
\leq& \frac{3\gamma^2 m^4 \left(1 + \ln m\right)^2 L_g^2 G_w^2}{\lambda^2 T_1^2},
\end{aligned}$$

the second step because, between iteration $s_j^{(t)}$ and iteration t we will perform $t - s_j^{(t)}$ updates of magnitude at most $G_w / \lambda T_1$, and the third step because, as an instance of the coupon collector's problem, $\max_j(t - s_j^{(t)})$ has expectation $mH_m \leq m + m \ln m$ (H_m is the mth harmonic number) and variance $m^2\pi^2/6$. Substituting into Equation 14:

$$\mathbb{E}\left[\tilde{h}\left(\bar{w}, p^*\right)\right] - f\left(w^*\right) \leq \frac{G_w^2\left(2 + \ln T_1 + \ln T_2\right)}{\lambda T_2} + \frac{4\gamma^2 L_g^2 \ln m}{\lambda T_2} + \frac{3m^4\left(1 + \ln m\right)^2 G_w^2}{2\lambda T_1^2}.$$

By the λ-strong convexity of $\tilde{h}$:

$$\mathbb{E}\left[\left\|\bar{w} - w^*\right\|_2^2\right] \leq \frac{2G_w^2\left(2 + \ln T_1 + \ln T_2\right)}{\lambda^2 T_2} + \frac{8\gamma^2 L_g^2 \ln m}{\lambda^2 T_2} + \frac{3m^4\left(1 + \ln m\right)^2 G_w^2}{\lambda^2 T_1^2}.$$

Using the facts that $\|\Pi_g(\bar{w}) - w^*\| \leq \|\bar{w} - w^*\|$ because w^* is feasible, and that $G_w = G_f + \gamma G_g$, completes the proof. ■

We now move on to the main result: a bound on the number of iterations (equivalently, the number of stochastic loss gradients) and constraint checks required to achieve ϵ-suboptimality:

Theorem 4 *Suppose that the conditions of Lemmas 1 and 16 apply, with the p-update step size η as defined in Lemma 16. If we run Algorithm 3 (MidTouch) for $T_{\epsilon 1}$ iterations in the first phase and $T_{\epsilon 2}$ in the second:*

$$\begin{aligned}T_{\epsilon 1} =& \tilde{O}\left(\frac{m\left(\ln m\right)^{2/3}\left(G_f+\gamma G_g+\gamma L_g\right)^{4/3}}{\lambda^{4/3}\epsilon^{2/3}}+\frac{m^2\left(\ln m\right)\left(G_f+\gamma G_g\right)}{\lambda\sqrt{\epsilon}}\right),\\ T_{\epsilon 2} =& \tilde{O}\left(\frac{\left(\ln m\right)\left(G_f+\gamma G_g+\gamma L_g\right)^2}{\lambda^2\epsilon}+\frac{m^{3/2}\left(\ln m\right)^{3/2}\left(G_f+\gamma G_g\right)^{3/2}}{\lambda^{3/2}\epsilon^{3/4}}\right),\end{aligned}$$

requiring a total of C_ϵ "constraint checks" (evaluations or differentiations of a single g_i):

$$\begin{aligned}C_\epsilon =& \tilde{O}\left(\frac{\left(\ln m\right)\left(G_f+\gamma G_g+\gamma L_g\right)^2}{\lambda^2\epsilon}+\frac{m^{3/2}\left(\ln m\right)^{3/2}\left(G_f+\gamma G_g\right)^{3/2}}{\lambda^{3/2}\epsilon^{3/4}}\right.\\ &\left.+\frac{m^2\left(\ln m\right)^{2/3}\left(G_f+\gamma G_g+\gamma L_g\right)^{4/3}}{\lambda^{4/3}\epsilon^{2/3}}+\frac{m^3\left(\ln m\right)\left(G_f+\gamma G_g\right)}{\lambda\sqrt{\epsilon}}\right),\end{aligned}$$

then:

$$\mathbb{E}\left[\left\|\Pi_g(\bar{w})-w^*\right\|_2^2\right]\le\mathbb{E}\left[\left\|\bar{w}-w^*\right\|_2^2\right]\le\epsilon,$$

where $w^*=\operatorname{argmin}_{\{w\in\mathcal{W}:\forall i.g_i(w)\le 0\}}f(w)$ is the optimal *constraint-satisfying reference vector.*

Proof We begin by introducing a number $\tau\in\mathbb{R}$ with $\tau\ge 1$ that will be used to define the iteration counts T_1 and T_2 as:

$$T_1=\left\lceil m\tau^2\right\rceil\qquad\text{and}\qquad T_2=\left\lceil\tau^3\right\rceil.$$

By Lemma 16, the above definitions imply that:

$$\begin{aligned}&\mathbb{E}\left[\left\|\Pi_g(\bar{w})-w^*\right\|_2^2\right]\\ &\quad\le\frac{2\left(G_f+\gamma G_g\right)^2\left(4+\ln m+5\ln\tau\right)+8\gamma^2L_g^2\ln m}{\lambda^2\tau^3}+\frac{3m^4\left(1+\ln m\right)^2\left(G_f+\gamma G_g\right)^2}{\lambda^2m^2\tau^4}\\ &\quad\le\frac{10\left(1+\ln m\right)\left(G_f+\gamma G_g+\gamma L_g\right)^2\left(1+\ln\tau\right)}{\lambda^2\tau^3}+\frac{3m^2\left(1+\ln m\right)^2\left(G_f+\gamma G_g\right)^2}{\lambda^2\tau^4}.\end{aligned}$$

Defining $\epsilon=\mathbb{E}\left[\left\|\Pi_g(\bar{w})-w^*\right\|_2^2\right]$ and rearranging:

$$\begin{aligned}&\lambda^2\epsilon\left(\frac{\tau}{\left(1+\ln\tau\right)^{1/3}}\right)^4\\ &\quad\le 10\left(1+\ln m\right)\left(G_f+\gamma G_g+\gamma L_g\right)^2\left(\frac{\tau}{\left(1+\ln\tau\right)^{1/3}}\right)+3m^2\left(1+\ln m\right)^2\left(G_f+\gamma G_g\right)^2.\end{aligned}$$

We will now upper-bound all roots of the above equation with a quantity τ_ϵ, for which all $\tau \geq \tau_\epsilon$ will result in ϵ-suboptimality. By the Fujiwara bound (Wikipedia, 2015), and including the constraint that $\tau \geq 1$:

$$\frac{\tau_\epsilon}{\left(1+\ln\tau_\epsilon\right)^{1/3}} \leq \max\left\{1, 2\left(\frac{10\left(1+\ln m\right)\left(G_f+\gamma G_g+\gamma L_g\right)^2}{\lambda^2\epsilon}\right)^{1/3}, \right. \\ \left. 2\left(\frac{3m^2\left(1+\ln m\right)^2\left(G_f+\gamma G_g\right)^2}{2\lambda^2\epsilon}\right)^{1/4}\right\}.$$

Substituting the above bound on τ_ϵ into the definitions of T_1 and T_2 gives the claimed magnitudes of these $T_{\epsilon 1}$ and $T_{\epsilon 2}$, and using the fact that the $C_\epsilon = O(mT_{\epsilon 1} + T_{\epsilon 2})$ gives the claimed bound on C_ϵ. ■