Safe Convex Learning under Uncertain Constraints

Ilnura Usmanova
Automatic Control Laboratory,
ETH Zürich, Switzerland

Andreas Krause
Machine Learning Institute,
ETH Zürich, Switzerland

Maryam Kamgarpour
Automatic Control Laboratory,
ETH Zürich, Switzerland

Abstract

We address the problem of minimizing a convex smooth function $f(x)$ over a compact polyhedral set D given a stochastic zeroth-order constraint feedback model. This problem arises in safety-critical machine learning applications, such as personalized medicine and robotics. In such cases, one needs to ensure constraints are satisfied while exploring the decision space to find optimum of the loss function. We propose a new variant of the Frank-Wolfe algorithm, which applies to the case of uncertain linear constraints. Using robust optimization, we provide the convergence rate of the algorithm while guaranteeing feasibility of all iterates, with high probability.[1]

1 INTRODUCTION

Many optimization tasks in robotics, health sciences, and finance require minimizing a loss function under uncertainties. Most existing stochastic and online optimization approaches proposed to address these tasks assume that the constraints of the corresponding optimization problems are known. These approaches, however, are unacceptable in cases in which the feasible set is itself unknown and is learned online. Optimizing a loss function under such a partially revealed feasible set model is further challenged by the fact that exploration can be made only inside the feasible set due to safety reasons. Hence, one needs to carefully choose actions to ensure feasibility of each iterate with high probability while learning the optimal solution. In the machine learning community, this problem is known as *safe learning*.

Safe learning is receiving increasing attention due to the increasingly widespread deployment of machine learning in safety-critical tasks. An example arises in personalized medicine, where physicians may choose from a large set of therapies. The effects of different therapies on the patient are initially unknown, and can only be determined through clinical trials. Free exploration however is not possible since some therapies might cause discomfort or even physical harm (Sui et al., 2015). Similar challenges arise in designing control algorithms for robots, which have to navigate unexplored terrains or interact with humans (Cassandra et al., 1996; Koenig and Simmons, 1996). In these scenarios, robots need to learn the best tuning for their controllers or optimize their trajectories based on risky experimental interactions with the partially known environment.

We address the problem of safe learning given a convex loss function subject to unknown constraints. Motivated by the aforementioned applications, we assume that the decision maker has access only to noisy observations of the constraints for a chosen action. Our objective is to design an algorithm that sequentially steers the decisions towards the optimum while ensuring safety of the decisions at every step. In this paper, we restrict ourselves to unknown linear constraints. Perhaps surprisingly, there is very little work on safe learning for this rudimentary setup. Hence, we consider this as a first step towards developing fundamental understanding and design of efficient algorithms for the more general nonlinear and non-convex setting.

Related work. There are many optimization algorithms that ensure feasibility of the iterates, assuming a known constraint function. The most basic ones are projected gradient descent (PGD)(Boyd and Vandenberghe, 2004) and Frank-Wolfe (FW) (Frank and Wolfe, 1956) (also known as conditional gradient). These methods require exact knowledge of the constraints or at

[1] We thank the support of Swiss National Science Foundation, under the grant SNSF 200021_172781, and ERC under the European Union's Horizon 2020 research and innovation programme grant agreement No 815943.

Proceedings of the 22$^{\text{nd}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

least a projection oracle or an exact linear programming (LP) oracle with respect to the constraints. However, as discussed above, this information may not be available in safe learning problems.

Assuming the functional form of the constraints, with parameters drawn from a probability distribution, chance-constraint optimization addresses the problem of optimizing a loss function subject to constraint satisfaction with a sufficiently high probability (El Ghaoui et al., 1998). The proposed solution methods for this problem assume either a priori knowledge of the distribution of the parameters (El Ghaoui et al., 1998), moments of the distribution (Zymler et al., 2013), or a sufficiently large number of samples of the distribution (Calafiore and Campi, 2005). In contrast, in a safe learning problem, the decision maker does not have access to such information a priori. This information is gathered online and feasibility needs too be ensured while exploring the uncertain decision space.

A recent line of work addresses uncertain constraints in online stochastic optimization (Yu et al., 2017; Yu and Neely, 2016). The work is based on infeasible penalty methods, and thus does not provide guarantees on constraint violation at each iteration. Rather, the methods ensure convergence of the *average* constraint violation to zero. Similarly, risk-aware contextual bandits and bandits with knapsack constraints (Sun et al., 2017; Mahdavi et al., 2012; Jenatton et al., 2015) consider unknown constraint functions with a budget limit. Here, safety refers to ensuring that the total usage of a commodity, e.g., budget for adverts, summed over the sequence of iterates remains below a threshold. Similar to Yu et al. (2017); Yu and Neely (2016), the above approaches bound average constraint violation, rather than avoiding violation at each iteration. While such a formulation can be well-suited in certain problems such as adverts, it may not be well-suited for safe learning applications discussed above because in this latter case, constraints need to be satisfied at each step.

The problem of safe learning using Gaussian processes (GP) has been proposed in Sui et al. (2015). The SafeOpt algorithm developed in the above work considers minimizing an unknown loss function iteratively, while ensuring that the loss of each iterate is above a required threshold. Given actively taken measurements of the loss, the initial estimate of the feasible set is incrementally enlarged through exploration and considering certain regularities of GP kernels. This framework is extended to multiple constraints and experimentally validated on robotic platforms by Berkenkamp et al. (2016). Safe GP learning is powerful as it can address general non-convex problems. Nevertheless, due to this generality, current approaches do not scale well with the problem dimension. This motivates our work of developing efficient safe learning algorithms for the case of convex loss functions and constraints.

Our contributions. We propose a novel algorithm for safe active learning, given a smooth convex objective and a set of unknown linear constraints with noisy oracle information. Given a confidence level $1-\delta$ and accuracy ϵ, we prove that after $\tilde{O}\left(\frac{1}{\epsilon}\right)$ iterations and $\tilde{O}\left(\frac{d^2 \ln 1/\delta}{\epsilon^2}\right)$ constraints measurements, the final point is an ϵ-accurate solution with probability $1-\delta$ (cf. Theorem 2). By $\tilde{O}(\cdot)$ we denote $O(\cdot)$ up to a logarithmic multiplicative factor. Furthermore, we ensure feasibility for the trajectory of the iterates with probability at least $1-\delta$ (cf. Theorem 1). While in this paper we mainly focus on exact first-order oracles for the objective function, we discuss extensions to stochastic oracles in Section 5.

The core idea of our algorithm is to combine a first-order feasible optimization approach with the robust optimization technique. Specifically, our algorithm is based on the Frank-Wolfe (FW) method. In each iteration, it solves an *uncertain* linear program based on estimates of the constraints and uses this solution to define the step direction. The safety of the iterates is ensured with high probability by refining the confidence set of the unknown parameters iteratively. We emphasize that while we use robust optimization (Ben-Tal and Nemirovski, 1998, 1999, 2000), our problem formulation is different than that of a classical robust optimization. Specifically, we consider gathering information online about the uncertainty, whereas the robust optimization works assume one-shot knowledge of uncertainties. We numerically evaluate the performance of the proposed algorithm in Section 6 and compare its performance with a one-shot robust optimization approach.

Notations Let $I^d \in \mathbb{R}^{d\times d}$ denote identity matrix, $e_i \in \mathbb{R}^d$ the unit vector corresponding to the i-th coordinate. Let $\|\cdot\|$ denote the Euclidean norm for vectors and the spectral norm for matrices. The ball of radius r centered at a point $x_0 \in \mathbb{R}^d$ is denoted by $B^d(x_0, r) = \{x \in \mathbb{R}^d | \|x - x_0\| \leq r\}$.

2 PROBLEM FORMULATION

The problem of safe learning in its most general form can be defined as a constrained optimization problem

$$\begin{aligned} \min_{x\in\mathbb{R}^d} \quad & f(x) \\ \text{subject to } & f_i(x) \leq 0 \quad \forall i = 1,\ldots,m, \end{aligned}$$

where the objective function $f : \mathbb{R}^d \to \mathbb{R}$ and the constraints $f_i : \mathbb{R}^d \to \mathbb{R}$ are unknown, and can only be accessed at feasible points x. A possibly noisy oracle provides access to the values of these functions or their gradients at any queried feasible $x \in \mathbb{R}^d$. The objective is to design an iterative algorithm that chooses the query points to ensure feasibility at each round while progressing towards the optimum. To address this goal, we need to define the oracle more precisely and make some regularity assumptions on the functions.

In this paper, we consider an instance of the safe learning problem in which the objective f is convex and M-Lipschitz continuous, that is, $|f(x) - f(y)| \leq M\|x - y\|$ $\forall x, y \in D$, where D is the feasible set. Furthermore, we assume f is L-smooth, that is, f has L-Lipschitz continuous gradients in D, $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$, $\forall x, y \in D$. We assume access to the gradients of the objective function, $\nabla f(x)$, at any feasible query point $x \in D$. We furthermore assume that constraints are known to be linear, $f_i(x) = [a^i]^T x - b^i$ for $i = 1, \ldots, m$. Hence, letting $A \in \mathbb{R}^{m \times d}$ denote the matrix with rows defined by $[a^i]^T$, the problem is given by

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & f(x) \\ \text{subject to} \quad & Ax - b \leq 0. \end{aligned} \tag{1}$$

We assume that the feasible set $D = \{x \in \mathbb{R}^d : Ax - b \leq 0\}$ is a compact polytope with non-empty interior. Denote by Γ the diameter of the set D, $\Gamma = \max_{x,y \in D} \|x - y\|$. Furthermore, let Γ_0 be the radius of the smallest ball centered at 0 such that $D \in B(0, \Gamma_0)$, namely, $\Gamma_0 = \max_{x \in D} \|x\|$.

If A and b are known, (1) can be solved efficiently by off-the-shelf first-order convex optimization algorithms. We however, consider the case in which A and b are unknown and can be accessed through an oracle. Specifically, we assume the constraints can be evaluated at any point that lies within a ball of radius ω_o of the feasible set. These evaluations are corrupted by Gaussian noise. Hence, we have access to $y(x) = Ax - b + \eta$ for any x such that $B^d(x, \omega_0) \cap D \neq \emptyset$, where $\eta = N(0, \sigma^2 I^m)$. If in the problem setting having all the measurements inside the feasible set is critical, we can artificially shrink the set D by the value ω_0 from the boundaries. This can be achieved by tightening the constraints $[a^i]^T x \leq b^i$ with setting the measurements $\hat{y}^i = y^i - \kappa = [a^i]^T x - b^i + \eta - \kappa^i$, with $\kappa^i \geq L_A^i \omega_0$, where L_A^i is an upper bound on $\|a^i\|$.

The scope of the present paper is to design an algorithm which, starting from a feasible point $x_0 \in D$, converges to an optimal solution x_* with a required accuracy ϵ and a required confidence $1 - \delta$ after T steps, that is,

$$\mathbb{P}\{f(x_T) - f(x_*) \leq \epsilon\} \geq 1 - \delta. \tag{2}$$

Since the constraint set D is unknown and revealed through a noisy oracle, we can at the very best ensure to remain inside the feasible set with sufficiently high probability. Hence, we require that the updates of the method are not violating the true constraints with the same required confidence level of $1 - \delta$, that is,

$$\mathbb{P}\{Ax_t - b \leq 0,\ 0 \leq t \leq T\} \geq 1 - \delta. \tag{3}$$

Some words on the choices of the optimization and oracle above are in order. First, the setting of linear constraints can be restrictive for some real-world problems. Nevertheless, understanding the linear setup is often the first step in addressing more challenging formulations. Second, having a noisy first-order or a zeroth-order oracle for the objective function is more realistic for several safe learning problems. Optimization under such oracle models have been deeply explored for the case in which the constraint set is known. Hence, the main novelty and challenge in safe learning is ensuring feasibility of the iterates despite uncertain and incrementally revealed constraint values. We discuss how the proposed algorithm can be generalized to stochastic oracle models for objective in Section 5.

3 THE SFW ALGORITHM

We propose a variant of the Frank-Wolfe algorithm where we explicitly take into account the uncertainty about the feasible set D, referred to as Safe Frank-Wolfe (SFW). The algorithm can be summarized as follows. Starting with a feasible point $x_0 \in D$, at each iteration $t = 0, \ldots, T$ we generate a number n_t of query points and obtain noisy measurements of the constraint functions at these points. Using linear regression, we obtain an estimate $\hat{D}_t$ of the feasible set based on the history of obtained measurements. The algorithm then uses $\hat{D}_t$ to obtain a direction $\hat{s}_t$ by solving the estimated Direction Finding Subproblem (DFS)

$$\hat{s}_t = \arg\min_{s \in \hat{D}_t} \langle \nabla f(x_t), s \rangle. \tag{4}$$

The next iterate is then given by $x_{t+1} = x_t + \gamma_t(\hat{s}_t - x_t)$, according to a chosen step-size γ_t. Below, we further describe each step of the proposed algorithm.

Taking Measurements. During each iteration t of the algorithm, we first make measurements at n_t number of points $x_{(j)}$ within distance ω_0 of x_t in d linearly independent directions. The number n_t needs to satisfy a lower bound as a function of the input data δ, T, to ensure safety. This bound is provided in Theorem 1. Denote by $X_t = [x_{(1)}, \ldots, x_{(N_t)}]^T \in \mathbb{R}^{N_t \times d}$ and by $N_t = \sum_{k=0}^{t} n_k$, the total number of available measurements at iteration t. Combining all measurements taken up to iteration t we have the following information about the constraints $y^i = X_t a^i - b^i \mathbf{1} + \eta^i$, $i = 1, \ldots, m$, where $y^i \in \mathbb{R}^{N_t}$ is the vector of N_t measurements of i-th constraint, $\eta^i \in \mathbb{R}^{N_t}$ is the vector of errors distributed according to the Gaussian distribution $N(0, \sigma^2 I^{N_t})$, and $\mathbf{1} \in \mathbb{R}^{N_t}$ the vector of 1's. Let us denote by $Y_t = [y^1, \dots, y^m] \in \mathbb{R}^{N_t \times m}$ the matrix of corresponding measurements of the constraints.

Estimating Constraints. Let $\beta^i = \left[[a^i]^T \quad b^i\right]^T \in \mathbb{R}^{d+1}$ denote the vector corresponding to the i-th constraint. We refer to β^i as the true parameter. Let $\bar{X}_t = [X_t, -\mathbf{1}] \in \mathbb{R}^{N_t \times (d+1)}$ be the extended version of the matrix X_t. The Least Squares Estimation (LSE) of the constraint parameters at step t is given by

$$\hat{\beta}_t = [\hat{A}_t, \hat{b}_t]^T = [\bar{X}_t^T \bar{X}_t]^{-1} \bar{X}_t^T Y_t. \tag{5}$$

The covariance matrix of the $\hat{\beta}_t^i$ is given by $\Sigma_t = \sigma^2 [\bar{X}_t^T \bar{X}_t]^{-1}$. Let $\hat{a}_t^i, \hat{b}_t^i$ denote the estimates of the corresponding rows of $\hat{\beta}_t^i$ and $\hat{D}_t = \{x \in \mathbb{R}^d : \hat{A}_t x \leq \hat{b}_t\}$ denote the estimated feasible set.

Stopping criteria. Recall that $\hat{s}_t$ is the minimizer of the estimated DFS (4) and let $\hat{g}_t$ be its optimal value

$$\hat{g}_t = \min_{s \in \hat{D}_t} \langle s, \nabla f(x_t) \rangle. \tag{6}$$

Similarly, let s_t denote the minimizer of the DFS under true constraints and g_t the corresponding optimal value

$$s_t = \arg\min_{s \in D} \langle s, \nabla f(x_t) \rangle, \quad g_t = \min_{s \in D} \langle s, \nabla f(x_t) \rangle. \tag{7}$$

From convexity of f, we have $f(x_t) - f(x_*) \leq g_t$. Thus, as discussed in Jaggi (2013), g_t can be taken as a surrogate duality gap and consequently a stopping criterion for the FW algorithm. In our case, the duality gap cannot be computed exactly because the feasible set D is unknown. Nevertheless, for the random variable $E_t := |\hat{g}_t - g_t|$ describing an error in the gap estimation we can derive a probabilistic upper bound $E_t(\delta)$, such that $\mathbb{P}\{E_t \leq E_t(\delta)\} \geq 1 - \delta$ (see Proposition 1 Section 5). It follows that if $\hat{g}_t + E_t(\delta) \leq \epsilon$, then with probability greater than $1 - \delta$ we have $f(x_t) - f(x_*) \leq \epsilon$. Thus, we use $\hat{g}_t + E_t(\delta) \leq \epsilon$ as a stopping criterion.

Putting the above few steps together, we present the Safe Frank-Wolfe (SFW) in Algorithm 1.

4 SAFETY

In order to ensure safety of the trajectory as per Inequality (3) we ensure that each x_{t+1} generated by the algorithm above remains within the feasible set D with probability $1 - \bar{\delta}$, where $\bar{\delta} = \frac{\delta}{T}$. This is achieved using the analysis framework of robust optimization by Bertsimas et al. (2011), Ben-Tal and Nemirovski (1998). The safety of each iterate, combined with a union bound, enables us to prove the safety of the sequence $\{x_t\}_{t=1}^T$ with probability $1 - \delta = 1 - \sum_{t=1}^T \bar{\delta}$.

Algorithm 1 SFW (Safe Frank-Wolfe)

```
Input: $x_0 \in D$, bound on iterations $T$, accuracy $\epsilon$, confidence parameter $\delta$, measurement radius $\omega_0$;
$t \leftarrow 0$; Choose $n_t(\delta, T)$;
while $t \leq T$ do
    Pick $2d$ points around the current point $x_t$
        $x_{(N_{t-1}+i)} = x_t + e_i\omega_0,$
        $x_{(N_{t-1}+2i)} = x_t - e_i\omega_0, i = 1, \dots, d,$
    and take $[n_t/2d]$ measurements at each point;
    Obtain the gradient $\nabla f(x_t)$ and the noisy constraint values $y_{(j)}^i = x_{(j)}^T a^i - b^i + \eta_{(j)}^i$ $\forall j = N_t + 1, \dots, N_t + n_t, i = 1, \dots, m$;
    Compute the LSE of the constraints $\hat{A}_t$ and $\hat{b}_t$ based on (5);
    Solve the estimated DFS (4) to obtain $\hat{s}_t$;
    Estimate the duality gap $\hat{g}_t$ (6);
    if $\hat{g}_t \leq \epsilon - E_t(\bar{\delta})$ then break and return $x_t$;
    Set $\gamma_t = \frac{1}{t+2}$;
    $x_{t+1} \leftarrow x_t + \gamma_t(\hat{s}_t - x_t)$;
    $t \leftarrow t + 1$.
```

Since the LSE's of the constraint parameters $\hat{\beta}_t^i = [[\hat{a}_t^i]^T, \hat{b}_t^i]^T \in \mathbb{R}^{d+1}$ are Gaussian with mean $\mathbb{E}\hat{\beta}_t^i = \beta^i$ and covariance matrix $\Sigma_t = \sigma^2 [\bar{X}_t^T \bar{X}_t]^{-1}$, the confidence set for the vector of true parameters β^i is given by the following ellipsoid (Draper and Smith, 2014): $\mathcal{E}_t^i(\bar{\delta}) = \left\{ z \in R^{d+1} : (\hat{\beta}_t^i - z)^T \Sigma_t^{-1} (\hat{\beta}_t^i - z) \leq \phi^{-1}(\bar{\delta})^2 \right\}$, where, $\phi^{-1}(\bar{\delta})^2$ denotes the inverse of Chi-squared cumulative distribution function with $d+1$ degrees of freedom. Thus, we have an ellipsoidal uncertainty set centered at $\hat{\beta}_t^i$, such that $\mathbb{P}\{\beta^i \in \mathcal{E}_t^i(\bar{\delta})\} \geq 1 - \bar{\delta}$. We define the confidence set $\mathcal{E}_t(\bar{\delta})$ for all parameters β by $\mathcal{E}_t(\bar{\delta}) = \mathcal{E}_t^1(\bar{\delta}/m) \times \ldots \times \mathcal{E}_t^m(\bar{\delta}/m) \subseteq \mathbb{R}^{(d+1)m}$. The confidence set $\mathcal{E}_t(\bar{\delta})$ determines the uncertainty set for constraint parameters β with probability $1 - \bar{\delta}$. Indeed, $1 - \mathbb{P}(\exists i : \beta^i \notin \mathcal{E}_t^i(\bar{\delta}/m)) = 1 - \mathbb{P}\{\cup_{i=1}^m \{\beta^i \notin \mathcal{E}_t^i(\bar{\delta}/m)\}\} \geq 1 - \sum_{i=1}^m \bar{\delta}/m = 1 - \bar{\delta}$.

We define the safety set $S_t(\bar{\delta}) \subset \mathbb{R}^d$ at iteration t as the set of $x \in \mathbb{R}^d$ satisfying the constraints with any true parameter $\beta = [A, b]$ in the confidence set:

$$S_t(\bar{\delta}) = \{x \in \mathbb{R}^d : Ax \leq b, \ \forall [A, b] \in \mathcal{E}_t(\bar{\delta})\}. \tag{8}$$

Given that for each constraint β^i our uncertainty set $\mathcal{E}_t^i(\bar{\delta})$ has an ellipsoidal form, the safety set $S_t(\bar{\delta})$ is equivalent to the intersection of a set of second order cone constraints (Ben-Tal et al., 2009)

$$S_t(\bar{\delta}) = \Bigg\{x \in\mathbb{R}^d : \forall i = 1,\dots,m \ \left[[\hat{a}^i_t]^T x - \hat{b}^i_t\right] + \\ + \phi^{-1}(\bar{\delta}/m) \left\| \Sigma_t^{1/2} \begin{bmatrix} x \\ -1 \end{bmatrix} \right\| \le 0 \Bigg\}. \quad (9)$$

Fact 1. *From the definition of the confidence and safety sets it readily follows that*

$$\mathbb{P}\{x \in D \,|\, x \in S_t(\bar{\delta}), \beta \in \mathcal{E}_t(\bar{\delta})\} = 1.$$

Fact 2. *The condition $x_t \in S_t(\bar{\delta})$ is equivalent to*

$$\phi_{\bar{\delta}} \sqrt{\frac{1}{N_t} + (x_t - \bar{x}_t)^T R_t (x_t - \bar{x}_t)} \le \min_{i=1,\dots,m} \varepsilon^i_t,$$

where $\phi_{\bar{\delta}} = \sigma\phi^{-1}(\bar{\delta}/m)$, $\varepsilon^i_t = \hat{b}^i_t - [\hat{a}^i_t]^T x_t$, $\bar{x}_t = \frac{X_t^T \mathbf{1}}{N_t}$, and $R_t = \left[\sum_{j=1}^{N_t} (x_{(j)} - \bar{x}_t)(x_{(j)} - \bar{x}_t)^T\right]^{-1}$.

The proof of this fact is provided in Appendix A.

To state the main result on safety of each iteration, we need to introduce some notation. For the polytope $D \in \mathbb{R}^d$, by an active set B we denote a set of indices of d linearly independent constraints active in a vertex $V \in \mathbb{R}^d$ of D, i.e., $V = V^B = [A^B]^{-1} b^B$. Here, A^B is a corresponding sub-matrix of A and b^B is the corresponding right-hand-side of the constraint. Let $\rho_{\min}(A^B)$ denote the smallest singular value of A^B. Let $Act(D)$ denote the set of all active sets corresponding to vertices of D, i.e., $Act(D) = \{B : V^B \text{ is a vertex of } D\}$. Furthermore, define $\rho_{\min}(D) := \min\{\rho_{\min}(A^B) : B \in Act(D)\}$. Note that $\rho_{\min}(D) > 0$ since by definition, B is a set of linearly independent active constraints. Let $\varepsilon_0 = \min_i\{b^i - [a^i]^T x_0\}$, and $L_A = \max_i \|a^i\|$. With the notation in place, we can present the following lemma on the lower bound on the number of measurements to ensure safety of each iterate.

Lemma 1. *If $\beta \in \mathcal{E}_k(\bar{\delta})$ for $k = 1,\dots,m$ and $n_t = 4C_n t (\ln t)^2$, with the constant parameter C_n satisfying*

$$C_n \ge C_{\bar{\delta}}^2 \max\left\{\frac{4(\ln\ln T)^2 L_A^2}{[\varepsilon_0]^2}, \frac{1}{(\Gamma_0+1)^2}\right\}, \quad (10)$$

$$\text{where } C_{\bar{\delta}} = \frac{2\phi_{\bar{\delta}} d(\Gamma_0+1)}{\rho_{\min}(D)} \sqrt{\frac{\Gamma_0^2+1}{\omega_0^2} + 1}, \quad (11)$$

then $x_t \in S_t(\bar{\delta})$. Furthermore, the total number of measurements then satisfies $N_t = C_n t^2 (\ln t)^2$.

We provide the full proof in Appendix C.

Proof sketch. Let us give a brief intuition for the proof. First, from Fact 2 we see that in order to have $x_t \in S_t(\bar{\delta})$ we require $\min_i \varepsilon^i_t \ge \Omega\left(\frac{1}{\sqrt{N_t}}\right)$, where ε^i_t is equal to the distance to the boundary corresponding to the estimated i-constraint multiplied by $\|\hat{a}^i_t\|$. Second, if these estimates were fixed, then using step sizes $\gamma_t = \frac{1}{t+2}$ we could ensure that the convergence to any boundary i would not be faster than $\varepsilon^i_t \ge \prod_{k=0}^t (1-\gamma_k)\varepsilon^i_0 = \frac{\varepsilon^i_0}{t+2}$ (see Figure 1). Hence, we require $N_t \ge \Omega\left(\frac{t^2}{\varepsilon_0^2}\right)$. However, since ε^i_t are random and boundaries are fluctuating, we need N_t to be square logarithmic times larger than the above estimate (as shown in the full proof). ■

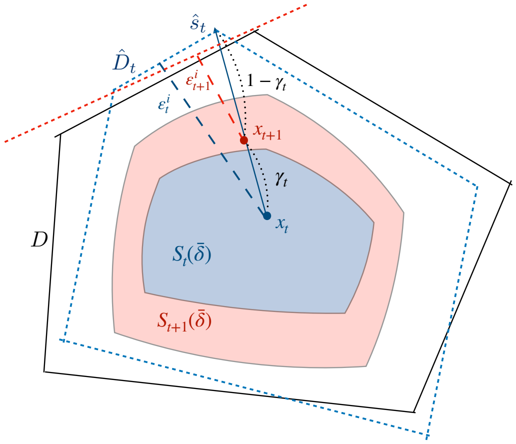


Figure 1: Illustration of one iteration of SFW $x_{t+1} = x_t + \gamma_t(\hat{s}_t - x_t)$. Bold lines denote the polytope D and dashed lines denote its estimate $\hat{D}_t$.

Remark. Note that the dependence on $\ln\ln T$ is very mild because this term grows extremely slowly, i.e, $\ln\ln 15 \approx 1$ and $\ln\ln 2000 \approx 2$.

Having established Lemma 1, we can present the safety guarantee of the SFW algorithm.

Theorem 1. *If $n_t = 4C_n t(\ln t)^2$, where the constant parameter C_n is defined in (10) then, the sequence of iterates $\{x_t\}_{t=0}^T$ of SFW is feasible with probability at least $1-\delta$, that is, $\mathbb{P}\{x_t \in D$ for all $0 \le t \le T\} \ge 1-\delta$.*

Proof. Let $\mathcal{F}_t$ denote the event that all the estimated confidence sets $\mathcal{E}_k(\bar{\delta})$ up to step t cover β, i.e., $\mathcal{F}_t = \{\beta \in \cap_{k=0}^t \mathcal{E}_k(\bar{\delta})\}$. Furthermore, let $\mathcal{Q}_t$ denote the event that all the $x_k \in S_k(\bar{\delta})$ up to iteration t, i.e, $\mathcal{Q}_t = \{x_k \in S_k(\bar{\delta})$ for all $0 \le k \le t\}$. By Lemma 1 if $\mathcal{F}_t$ holds, then $x_t \in S_t(\bar{\delta})$. Hence, it is easy to see that $\mathcal{F}_t$ implies $\mathcal{Q}_t$, i.e., $\mathbb{P}\{\mathcal{Q}_t|\mathcal{F}_t\} = 1$. Thus, using Fact 1 we derive

$$\mathbb{P}\left\{x_t \in D \text{ for all } 0 \le t \le T\right\} = \mathbb{P}\{\mathcal{Q}_T, \mathcal{F}_T\} = \mathbb{P}\{\mathcal{F}_T\}.$$

Using Boole's inequality we can bound the probability of $\mathcal{F}_T$ as follows

$$\begin{aligned}\mathbb{P}\{\mathcal{F}_T\} &= 1 - \mathbb{P}\{\cup_{t=0}^{T} \cup_{i=1}^{m} \{\beta^i \notin \mathcal{E}_t^i(\bar{\delta}/m)\}\} \\ &\geq 1 - \sum_{t=0}^{T}\sum_{i=1}^{m} \bar{\delta}/m \geq 1 - T\bar{\delta}.\end{aligned}$$

This concludes the proof. ■

5 CONVERGENCE

First, we show that the proposed algorithm achieves the optimal convergence rate for the Frank-Wolfe algorithm (see Lan (2013)), with sufficiently high probability. Second, we discuss extensions to stochastic first-order and zeroth-order oracles of the objective function based on the results of Hazan and Luo (2016).

5.1 Convergence rate.

Let us define the curvature constant C_f of the function $f(x)$ with respect to the compact domain D by

$$C_f = \sup_{\substack{x,s\in D,\gamma\in[0,1],\\ y=x+\gamma(s-x)}} \frac{1}{\gamma^2}(f(y) - f(x) - \langle y - x, \nabla f(x)\rangle).$$

It can be verified that $C_f \leq L\Gamma^2$, where L is the Lipschitz constant of the gradient $\nabla f(x)$ and Γ is the diameter of the set D (see Section 2). Our main result is as follows.

Theorem 2. *If $n_t = 4C_n(t+2)(\ln(t+2))^2$ and C_n is chosen according to the bound in Equation (10), then:*
a) after T steps of the SFW algorithm, the final point x_T satisfies

$$\mathbb{P}\left\{ f(x_T) - f(x_*) \leq \frac{f(x_0) - f(x_*)}{T+2} + \right. \\ \left. + \frac{\ln(T+2)\frac{C_f}{2} + \ln\ln(T+2)\frac{C'}{2}}{T+2} \right\} \geq 1 - \delta,$$

where $C' = \frac{MC_{\bar{\delta}}}{\sqrt{C_n}}$, and $C_{\bar{\delta}}$ is defined in (11).
b) all the iterates $\{x_t\}_{t=1}^{T}$ are feasible with probability $1-\delta$, as required in (3).

Corollary 1. *The SFW algorithm achieves an ϵ-accurate solution with probability greater than $1-\delta$ after making $\tilde{O}(\frac{1}{\epsilon})$ linear optimization oracle calls and $\tilde{O}\left(\frac{d^2 \ln\frac{1}{\delta}}{\epsilon^2}\right)$ zeroth-order inexact constraint oracle calls.*

Below, we provide the proof sketch for Theorem 2. The full proofs of Theorem 2 and Corollary 1 are provided in Appendix D.

Proof sketch. Our proof is based on the extensive study of FW convergence provided by Jaggi (2013), Freund and Grigas (2016). Recall that E_t is the accuracy with which an approximated DFS at iteration t is solved. Similarly to (Freund and Grigas (2016), Theorem 5.1) we can show that for $\gamma_t = O\left(\frac{1}{t}\right)$, we have

$$f(x_t) - f(x_*) \leq O\left(\frac{\epsilon_0 + C_f \ln t + \sum_{t=1}^{T} E_t}{t}\right), \quad (12)$$

where $\epsilon_0 = f(x_0) - f(x_*)$. Hence, to prove the convergence rate of the SFW algorithm we need to show that the error in the DFS solution decreases with the rate $O\left(\frac{1}{t}\right)$. This fact is shown in Proposition 1 below.

Proposition 1. *If $\beta \in \mathcal{E}_t(\bar{\delta})$ and $N_t \geq \frac{C_{\bar{\delta}}^2}{(\Gamma_0+1)^2}$, then $E_t \leq \frac{MC_{\bar{\delta}}}{\sqrt{N_t}}$. Hence, $\mathbb{P}\left\{E_t \leq \frac{MC_{\bar{\delta}}}{\sqrt{N_t}}\right\} \geq 1 - \bar{\delta}$.*

We provide the proof in Appendix B. From the result above it directly follows that $E_t = O\left(\frac{1}{t\ln t}\right)$, hence $\sum_{k=0}^{t} E_k = O(\ln\ln t)$. Using this result, and the classical FW proof technique (Freund and Grigas (2016)) we can conclude the result of Theorem 2. This concludes the proof sketch. ■

We can see that under our choice of the number of measurements n_t at each iteration, the total number of measurements is $O\left(d^2t^2\ln t^2\right)$. It follows that the required number of measurements at each step grows almost linearly with the iteration number and quadratically with the dimension d. Note however that the number of iterations is independent of the dimension d. In contrast, the safe learning approach in (Sui et al., 2015) is based on gridding the decision space and hence, the dependence in d is exponential. Hence, compared to previous safe learning approaches (Sui et al., 2015; Berkenkamp et al., 2016), our method scales better with dimension. Naturally, this scalability is due to the assumption of convexity of the cost function and the linearity of the constraints.

Finally, let us clarify some computational complexity aspects. After adding each new data point to X, the matrix inversion $(X^TX)^{-1}$ can be performed using one-rank updates (e.g., using formula $(A+vv^T)^{-1} = A^{-1} - \frac{1}{(1+v^TA^{-1}v)}(A^{-1}vv^TA^{-1})$). The cost of each such operation is $O(d^2)$. This operation is to be made $N_t = O(d^2t^2(\ln t)^2)$ times. The total computational complexity is thus $O(d^2N_t) = O(d^4t^2(\ln t)^2)$ and additionally t LP oracle calls.

5.2 Extension to stochastic oracle for the objective function.

Notice that the SFW algorithm requires $t = \tilde{O}\left(\frac{1}{\epsilon}\right)$ iterations and $N_t = \tilde{O}\left(\frac{d^2\ln\frac{1}{\delta}}{\epsilon^2}\right)$ measurements of constraints to obtain a required accuracy of ϵ. General

Stochastic Frank-Wolfe algorithm with stochastic objective but known linear constraints require $t = O(\frac{1}{\epsilon})$ iterations and in contrast, $t = O(\frac{1}{\epsilon^3})$ stochastic gradient measurements (Hazan and Luo (2016), Table 2).[2] This difference in the number of measurements is due to the fact that in the absence of linearity of the objective function, the gradients of the objective function are changing in each iteration. Thus, $O(t^2)$ measurements at each iteration are needed to guarantee correct variance reduction rate of the Frank-Wolfe method (see (12)). From the above observation, we can extend the SFW analysis to the case in which we have access to a stochastic first-order oracle of the objective function. In this case, a total of $O(t^3)$ calls to the objective function oracle, and $O(t^2)$ calls to the constraints oracle are sufficient to obtain the desired rate of decrease of E_t in Proposition 1 and hence, the convergence rate in Theorem 2. Similarly, for the case of zeroth-order oracle $O(d^2t^3)$ calls are needed to estimate the gradient of the objective. The noisy gradient does not influence the safety of the proposed algorithm. Thus, the safety results in Theorem 1 extend to the case with stochastic first-order or zeroth-order oracle of the objective.

6 EXPERIMENTS

We evaluate the performance of the proposed approach experimentally. In the first experiment, we consider the convergence rate of the algorithm as a function of the dimension. In the second experiment, we compare the SFW algorithm with a robust optimization based approach, which first learns the uncertain constraints and then finds the optimum with respect to the estimated constraints. We consider the convex smooth optimization problem:

$$\min_{x \in D} \frac{1}{2} \left\| x - x' \right\|_2^2,$$

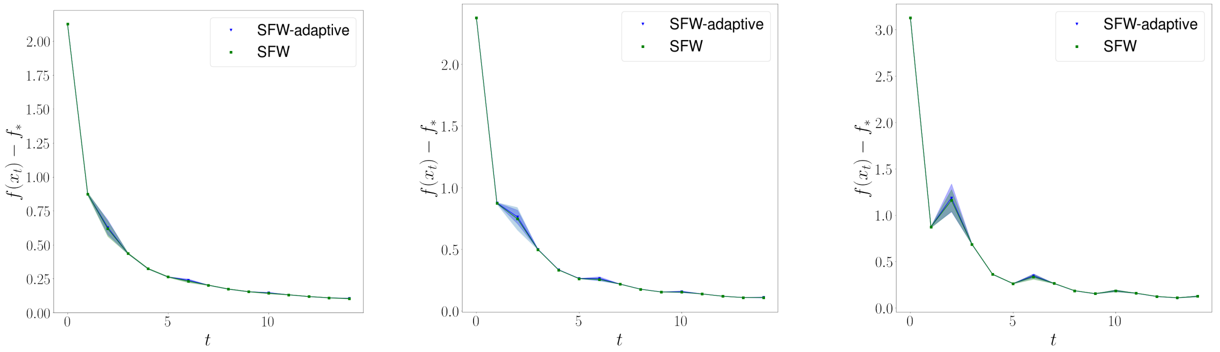
where $D = \left\{ x \in \mathbb{R}^d | -1 \leq x^i \leq 1, i = 1, \dots, d \right\}$ and $x' = [2, 0.5, \dots, 0.5] \in \mathbb{R}^d$ for varying dimension d. Then, the solution x_* is a point on the boundary of the true constraint set above. We set the variance of the noise to $\sigma = 0.01$ and use a constant exploration radius $\omega_0 = 0.01$. Furthermore, we set the confidence parameter $\delta = 0.1$ and the total number of iterations to $T = 15$.

Empirical constraint violation and convergence rate. The first experiment evaluates the empirical convergence rate and constraint violation as a function of the dimension d. First, we evaluate the convergence rate assuming we can obtain the required lower bound on C_n as per Theorem 2. We run the algorithm for dimensions $d = 2, 10, 20$. In particular, the parameters of the problem required for obtaining the lower bound are derived based on knowledge of the constraints and the objective function, as well as the input parameters δ, T as follows: $\varepsilon_0 = 1, \bar{\delta} = 0.0067, \phi_{\bar{\delta}} = 3.43$, $\|b\| = 2, \|a\| = 1$. It follows that a value of $C_n = d^2 \cdot 24$ achieves the required number of measurements. The SFW proposed in Algorithm 1 is then run with the above choices of parameters $\bar{\delta}$ and n_t. For each dimension, we run the SFW algorithm 20 times, keeping all the parameters and the initial conditions the same. The difference in each experiment is due to the stochastic noise in the measurements. The average and standard deviation of the function values $f(x_t) - f(x_*)$ scaled by the initial condition error are shown in Figure 2. It can be seen that the dimension does not influence the convergence rate, rather, it influences only the number of measurements.

We also run the experiment assuming we cannot compute the lower bound C_n precisely due to lack of problem data. In this case, at each iteration we first take $2dt$ measurements and further continuously take new measurements around x_t until the safety set $S_{t+1}(\bar{\delta})$ grows sufficiently to ensure x_{t+1} becomes safe (see Fact 2). This safety indeed verifies the feasibility of iterates with high probability based on Fact 1. Let us refer to this as the adaptive variant of SFW. This adaptive approach is not only more practical due to lack of dependence on problem data, but also requires far fewer measurements in total since the bound on n_t from the Theorem 1 is quite conservative. This is due to the fact that our theoretical bounds were derived using the worst-case estimate of $\|\Sigma_t^{1/2}\|$, when the measurements are taken always around the same point. However, $\|\Sigma_t^{1/2}\|$ can reduce much faster in practice. The convergence will also hold since the bound on $E_t \leq \frac{C_{\bar{\delta}} M}{\sqrt{N_t}} \leq \frac{C_{\bar{\delta}} M}{t+2} = O\left(\frac{1}{t}\right)$ required for the convergence rate is still satisfied. In Figure 2 caption we reported the required total number of measurements up to step T of the adaptive variant by N_T^a. You can see that it has significantly reduced compared to the non-adaptive variant.

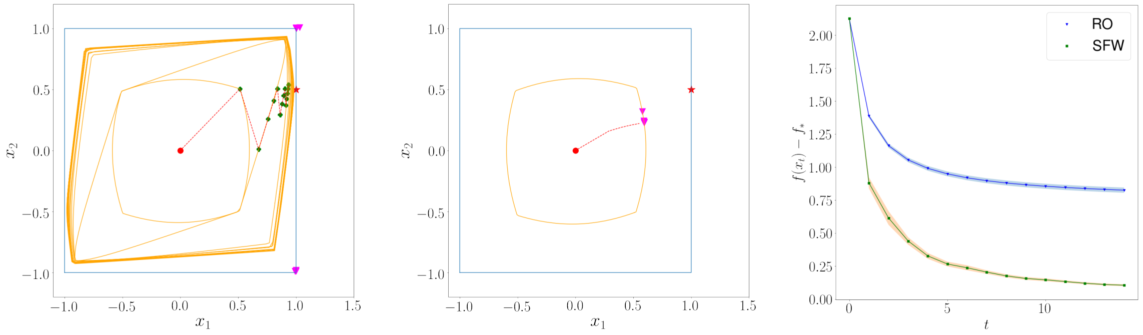
Comparison with an alternative robust optimization approach. We compare the proposed SFW method with an alternative approach in which we first make enough measurements in the a priori safe region to estimate the safety set (see Definition (8)) with sufficiently high probability. Next, we run a first-order method, such as FW with respect to the nonlinear set $S(\bar{\delta})$. Let us call this approach RO for robust optimization. To compare these two methods we set an a priori number of measurements for the alternative RO method equal to the total number of measurements N_t,

[2] The projection-free scheme for stochastic optimization proposed in (Lan and Zhou, 2016) achieves $N_t = O\left(1/\epsilon^2\right)$ measurements in total, but their method is much less straightforward.



(a) $d = 2, N_T = 8396, N_T^a = 519$ (b) $d = 4, N_T = 16792, N_T^a = 1135$ (c) $d = 10, N_T = 41980, N_T^a = 4275$

Figure 2: Convergence rate of SFW method for the dimensions $d = 2, 4, 10$ with $T = 15$.



(a) SFW (b) RO (c) Convergence rates of SFW and RO

Figure 3: Trajectories and accuracy of the objective value by each iteration of optimization. The left pair of plots shows the convergence of one realization of SFW method and of the robust optimization approach (without safety set updates) with $\sigma = 0.1$, $T = 15$, $N_T = 5500$. The orange lines denote the boundaries of the safety sets $S_t(\bar{\delta})$. Red circle denotes the starting point and star denotes the solution of the original problem. Magenta triangles denote the estimated DFS solutions.

of SFW algorithm corresponding to $\delta = 0.1$ and $T = 15$. After estimation of the safety set, we make $T = 15$ iterations of the FW method with the constraint set $S_T(\bar{\delta})$. Thus, the total number of measurement and the total number of optimization steps of the two methods are equal. During the run of the SFW, as per discussion in the above example, we reduce the number of measurements required to ensure safety at each iteration t, online. Figure 3(a),(b) shows the optimization trajectories of each method. The green round points along the trajectory correspond to the points where the constraints were measured. We also show the comparison of their convergence rates in Figure 3(c). As we can see, SFW algorithm performs better both in terms of estimates of the constraints and convergence rate. This difference in performance can be explained based on two observations. First, SFW moves measurements along the trajectory $\{x_t\}$, and this can lead to smaller variance of the estimates $\Sigma_t = \sigma^2 \left(\bar{X}_t^T \bar{X}_t\right)^{-1}$. Hence, the measurements are more informative and the safety set $S_T(\bar{\delta})$ is larger. Second, SFW algorithm is proven to converge to an ϵ-optimal solution corresponding to the true constraints. The RO approach however, can at the very best converge to an optimum with respect to a safety set estimated in advance. From the computational perspective, at each iteration the proposed SFW method requires an LP oracle, whereas the alternative RO approach requires solving a second-order cone program. Hence, the SFW is more tractable.

7 CONCLUSION

We proposed a safe learning approach for convex costs and uncertain linear constraints. This method uses information along the optimization trajectory to decrease the objective value and to explore an unknown feasible set. Meanwhile, it ensures feasibility for each iteration with high probability. We provided an analysis of convergence rate of our algorithm, as well as of feasibility guarantees for its iterations. Our next steps are to generalize the results to nonlinear constraints and to provide performance guarantees in terms of regret.

References

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.

Ben-Tal, A. and Nemirovski, A. (1998). Robust convex optimization. *Mathematics of operations research*, 23(4):769–805.

Ben-Tal, A. and Nemirovski, A. (1999). Robust solutions of uncertain linear programs. *Operations research letters*, 25(1):1–13.

Ben-Tal, A. and Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical programming*, 88(3):411–424.

Berkenkamp, F., Krause, A., and Schoellig, A. P. (2016). Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *arXiv preprint arXiv:1602.04450*.

Bertsimas, D., Brown, D. B., and Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM review*, 53(3):464–501.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Calafiore, G. and Campi, M. C. (2005). Uncertain convex programs: randomized solutions and confidence levels. *Mathematical Programming*, 102(1):25–46.

Cassandra, A. R., Kaelbling, L. P., Kurien, J., et al. (1996). Acting under uncertainty: discrete bayesian models for mobile-robot navigation. In *IROS*, volume 96, pages 963–972.

Draper, N. R. and Smith, H. (2014). *Applied regression analysis*, volume 326. John Wiley & Sons.

El Ghaoui, L., Oustry, F., and Lebret, H. (1998). Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9(1):33–52.

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

Freund, R. M. and Grigas, P. (2016). New analysis and results for the frank–wolfe method. *Mathematical Programming*, 155(1-2):199–230.

Hazan, E. and Luo, H. (2016). Variance-reduced and projection-free stochastic optimization. In *International Conference on Machine Learning*, pages 1263–1271.

Jaggi, M. (2013). Revisiting frank-wolfe: projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning-Volume 28*, pages I–427. JMLR. org.

Jenatton, R., Huang, J., and Archambeau, C. (2015). Adaptive algorithms for online convex optimization with long-term constraints. *arXiv preprint arXiv:1512.07422*.

Koenig, S. and Simmons, R. G. (1996). Unsupervised learning of probabilistic models for robot navigation. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 3, pages 2301–2308. IEEE.

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

Lan, G. and Zhou, Y. (2016). Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409.

Laurent, B. and Massart, P. (2000). Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338.

Mahdavi, M., Jin, R., and Yang, T. (2012). Trading regret for efficiency: online convex optimization with long term constraints. *Journal of Machine Learning Research*, 13(Sep):2503–2528.

Sui, Y., Gotovos, A., Burdick, J., and Krause, A. (2015). Safe exploration for optimization with gaussian processes. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 997–1005, Lille, France. PMLR.

Sun, W., Dey, D., and Kapoor, A. (2017). Safety-aware algorithms for adversarial contextual bandit. In *International Conference on Machine Learning*, pages 3280–3288.

Yu, H., Neely, M., and Wei, X. (2017). Online convex optimization with stochastic constraints. In *Advances in Neural Information Processing Systems*, pages 1427–1437.

Yu, H. and Neely, M. J. (2016). A low complexity algorithm with $o(\sqrt{T})$ regret and finite constraint violations for online convex optimization with long term constraints. *arXiv preprint arXiv:1604.02218*.

Zymler, S., Kuhn, D., and Rustem, B. (2013). Distributionally robust joint chance constraints with second-order moment information. *Mathematical Programming*, 137(1-2):167–198.