Safety-Aware Algorithms for Adversarial Contextual Bandit

Wen Sun [1] Debadeepta Dey [2] Ashish Kapoor [2]

Abstract

In this work we study the safe sequential decision making problem under the setting of adversarial contextual bandits with *sequential risk constraints*. At each round, nature prepares a context, a cost for each arm, and additionally a *risk* for each arm. The learner leverages the context to pull an arm and receives the corresponding cost and risk associated with the pulled arm. In addition to minimizing the cumulative cost, for safety purposes, the learner needs to make safe decisions such that the average of the cumulative risk from all pulled arms should not be larger than a pre-defined threshold. To address this problem, we first study online convex programming in the full information setting where in each round the learner receives an adversarial convex loss and a convex constraint. We develop a meta algorithm leveraging online mirror descent for the full information setting and then extend it to contextual bandit with sequential risk constraints setting using expert advice. Our algorithms can achieve near-optimal regret in terms of minimizing the total cost, while successfully maintaining a sublinear growth of accumulative risk constraint violation. We support our theoretical results by demonstrating our algorithm on a simple simulated robotics reactive control task.

1. Introduction

The topic of *Safe Sequential Decision Making* recently has received a lot of attention (Amodei et al., 2016) and finds its importance in different applications ranging from robotics, artificial intelligence and clinical trials. Designing reactive controls for mobile robots requires reasoning and making safe decisions so that robots can minimize the risk of dangerous obstacle collision. In clinical trials the risk of the side-effect of a new treatment must be taken into consideration for patients' safety. In general these applications will require the risk (probability of collision, level of side-effect) to be less than some safe-threshold. In this work we study safe sequential decision making under the setting of adversarial contextual bandits with sequential *risk constraints*.

The *Contextual Bandits* problem (Langford & Zhang, 2008) is a classic framework for studying sequential decision making with rich contextual information. In each round, given the contextual information, the learner chooses an arm (i.e., a decision) to pull based on the history of the interaction with the environment, and then receives the reward associated with the pulled arm. For the special case where contexts and rewards are i.i.d sampled from fixed unknown distributions, there exists an oracle-based computationally efficient algorithm (Agarwal et al., 2014) that achieves near-optimal regret rate. For adversarial contexts and rewards, EXP4 (Auer et al., 2002) and EXP4.P (Beygelzimer et al., 2011) are state-of-the-art algorithms, which achieve near-optimal regret rate, but are not computationally efficient in general. Recently, a few authors have started to incorporate global constraints into multi-armed bandit and contextual bandits problem where the goal of the learner is to maximize the reward while satisfying a global constraint to some degree. Previous work considered special cases such as single resource budget constraint (Ding et al., 2013; Madani et al., 2004) and multiple resources budget constraints (Badanidiyuru et al., 2013). Resourceful Contextual Bandits (Badanidiyuru et al., 2014) first introduced resource budget constraint to contextual bandits. Later on, the authors in (Agrawal et al., 2015) generalize the setting to a setting with a global convex constraint and a concave objective. Recently (Agrawal & Devanur, 2015) introduce a Upper Confidence Bound (UCB) style algorithm for linear contextual bandits with knapsack constraints. The settings considered in these previous work mainly focused on the stochastic case where contexts and rewards are i.i.d, and the single constraint is pre-fixed (i.e., time-independent, non-adversarial).

We introduce sequential risk constraints in contextual bandits: in each round, the environment prepares a context, a

[1] Robotics Institute, Carnegie Mellon University, USA [2] Microsoft Research, Redmond, USA. Correspondence to: Wen Sun <wensun@cs.cmu.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

They can be very computationally efficient for special cases of expert class (Beygelzimer et al., 2011) cost for each arm, and additionally a risk for each arm. The learner pulls an arm and receives the cost and risk associated with the pulled arm. Given a pre-defined risk threshold, the learner ideally needs to make safe decisions such that the risk is no larger than the safe-threshold in every round, while minimizing the cumulative cost simultaneously. Such adversarial risk functions are common in real world applications: when a robot is navigating in an unfamiliar environment, risk (e.g., probability of being collision with unseen obstacles) and reward of taking a particular action may dependent on the robot's current state and the environment (or the whole history of states), while the sequential states visited by the robot are unlikely to be i.i.d or even Markovian.

To address the adversarial contextual bandit with sequential risk constraints problem, we first study the problem of online convex programming (OCP) with sequential constraints, where at each round, the environment prepares a convex loss, and additionally a convex constraint. The learner wants to minimize its cumulative loss while satisfying the constraints as best as possible. The online learning with constraints setting is first studied in (Mannor et al., 2009) in a two-player game setting. Particularly the authors constructed a two-player game where there exists a strategy for the adversary such that among the strategies of the player that *satisfy the constraints on average*, there is no strategy that can achieve the no-regret property in terms of maximizing the player's reward. Later on (Mahdavi et al., 2012; Jenatton et al., 2016) considered the online convex programming framework where they introduced a pre-defined global constraint and designed algorithms that achieve no-regret property on loss functions while maintaining the accumulative constraint violation grows sublinearly. Though the work in (Mahdavi et al., 2012; Jenatton et al., 2016) did not consider time-dependent, adversarial constraints, we find that their online gradient descent (OGD) (Zinkevich, 2003) based algorithms are actually general enough to handle adversarial time-dependent constraints. We first present a family of online learning algorithms based on Mirror Descent (OMD) (Beck & Teboulle, 2003; Bubeck, 2015), which we show achieves near-optimal regret rate with respect to loss and maintains the growth of total constraint violation to be sublinear. With a specific design of a mirror map, our meta algorithm reveals a similar algorithm shown in (Mahdavi et al., 2012).

The mirror descent based algorithms in the full information online learning setting also enables us to derive a Multiplicative Weight (MW) update procedure by choosing negative entropy as the mirror map. Note that MW based update procedure is important when extending to partial information contextual bandit setting. The MW based update procedure can ensure the regret is polylogarithmic in the number of experts , instead of polynomial in the number of experts from using the OGD-based algorithms (Mahdavi et al., 2012; Jenatton et al., 2016). Leveraging the MW update procedure developed from the online learning setting, we present algorithms called EXP4.R (EXP4 with Risk Constraints) and EXP4.P.R (EXP4.P with Risk Constraints). EXP4.R can achieve near optimal regret in terms of minimizing cost while ensuring the average of the accumulative risk is no larger than the pre-defined threshold. For EXP4.P.R, we further introduce a tradeoff parameter that shows how one can trade between the risk violation and the regret of cost.

The rest of the paper is organized as follows. We introduce necessary definitions and problem setup in Sec. 2. We then deviate to the full information online learning setting where we introduce sequential, adversarial convex constraints in Sec. 3. In Sec. 4, we move to contextual bandits with risk constraints setting to present and analyze the EXP4.R and EXP4.P.R algorithm.

2. Preliminaries

2.1. Definitions

A function $R(x) : \mathcal{X} \to \mathcal{R}$ it is strongly convex with respect to some norm $\|\cdot\|$ if and only if there exists a constant $\alpha \in \mathcal{R}^+$ such that:

$$R(x) \geq R(x_0) + \nabla R(x_0)^T(x - x_0) + \frac{\alpha}{2}\|x - x_0\|^2.$$

Given a strongly convex function $R(\cdot)$, the Bregman divergence $D_R(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathcal{R}$ is defined as follows:

$$D_R(x, x') = R(x) - R(x') - \nabla R(x')^T(x - x').$$

2.2. Online Convex Programming with Constraints

In each round, the learner makes a decision $x_t \in \mathcal{X} \subseteq \mathcal{R}^d$, and then receives a convex loss function $\ell_t(\cdot)$ and a convex constraint in the form of $f_t(\cdot) \leq 0$. The learner suffers loss $\ell_t(x)$. The work in (Mahdavi et al., 2012) considers a similar setting but with a known, pre-defined global constraint. Instead of projecting the decision x back to the convex set induced by the global constraint $f(\cdot)$, (Mahdavi et al., 2012) introduces an algorithm that achieves no-regret on loss while satisfying the global constrain in a long-term perspective. Since exactly satisfying adversarial constraint in every round is impossible, we also consider constraint satisfaction in a long-term perspective. Formally, for the sequence of decisions $\{x_t\}_t$ made by the learner, we define $\sum_{t=1}^T f_t(x_t)$ as the cumulative constraint violation and we want to control the growth of the cumulative constraint violation to be sublinear: $\sum_{t=1}^T f_t(x_t) \in o(T)$, so that for the *long-term constraint* $\frac{1}{T}\sum_{t=1}^T f_t(x)$, we have $\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^T f_t(x_t) \leq 0$.

To be consistent to classic Online Convex Programming setting, we consider minimizing cost

We place one assumption on the decision set $\mathcal{X}$: we assume that the decision set $\mathcal{X}$ is rich enough such that in hindsight, we have $x \in \mathcal{X}$ that can satisfy all constraints: $\mathcal{O} \doteq \{x \in \mathcal{X} : f_t(x) \leq 0, \forall t\} \neq \emptyset$ (e.g., default conservative, safe decisions). We compete with the optimal decision $x^* \in \mathcal{O}$ that minimizes the total loss in hindsight:

$$x^* = \arg\min_{x \sim \mathcal{O}} \sum_{t=1}^{T} \ell_t(x). \tag{1}$$

Though one probably would be interested in competing against the best decision from the set of decisions that satisfy the constraints in average: $\mathcal{O}' \doteq \{x \in \mathcal{X} : (1/T)\sum_t^T f_t(x) \leq 0\}$, in general it is impossible to compete against the best decision in $\mathcal{O}'$ in hindsight ($\mathcal{O} \subset \mathcal{O}'$). The following proposition adapts the discrete 2-player game from proposition 4 in (Mannor et al., 2009) for the OCP with adversary constraints and shows the learner is unable to compete against $\mathcal{O}'$:

Proposition 2.1. *There exists a decision set $\mathcal{X}$, a sequence of convex loss functions $\{\ell_t(x)\}$, and a sequence of convex constraints $\{f_t(x) \leq 0\}$, such that for any sequence of decisions $\{x_1, ..., x_t, ...\}$, if it satisfies the long-term constrain as $\limsup_{t\to\infty} \frac{1}{t}\sum_{i=1}^{t} f_i(x_i) \leq 0$, then if competing against $\mathcal{O}'$, the regret grows at least linearly:*

$$\limsup_{t\to\infty} \Big(\sum_{i=1}^{t} \ell_i(x_i) - \min_{x\in\mathcal{O}'} \sum_{i=1}^{t} \ell_i(x)\Big) = \Omega(t). \tag{2}$$

The proof of the proposition can be found in Sec. A in Appendix. Hence in the rest of the paper, we have to restrict to $\mathcal{O}$. The average regret of loss R_ℓ and the average constraint violation R_f are defined as:

$$R_\ell = \frac{1}{T}[\sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(x^*)], \quad R_f = \frac{1}{T}[\sum_{t=1}^{T} f_t(x_t)].$$

We will assume the decision set is bounded as $\max_{x_1,x_2\in\mathcal{X}} D_R(x_1,x_2) \leq B \in \mathcal{R}^+$, $x \in \mathcal{X}$ is bounded as $\|x\| \leq X \in \mathcal{R}^+$, the loss function is bounded as $|\ell_t(\cdot)| \leq F \in \mathcal{R}^+$, the constraint is bounded $|f_t(\cdot)| \leq D \in \mathcal{R}^+$ and the gradient of the loss and constraint is also bounded as $\max\{\|\nabla_x \ell_t(x)\|_*, \|\nabla_x f_t(x)\|_*\} \leq G \in \mathcal{R}^+$, where $\|\cdot\|_*$ is the dual norm with respect to $\|\cdot\|$ defined for $\mathcal{X}$. Note the setting with a global constraint considered in (Mahdavi et al., 2012) is a special case of our setting. Set $f_t = f$, where f is the global constraint. If $R_f \in o(T)$, by Jensen's inequality, we have $f(\sum_{t=1}^{T} x_t/T) \leq \sum_{t=1}^{T} f(x_t)/T = o(T)/T \to 0$, as $T \to \infty$.

2.3. Contextual Bandits with Risk Constraints

For contextual bandits with sequential risk constraints, let $[K]$ be a finite set of K arms, $\mathcal{S}$ be the space of contexts. Formally, at every time step t, the environment generates a context $s_t \in \mathcal{S}$, a K-dimensional cost vector $c_t \in [0,1]^K$, and a risk vector $r_t \in [0,1]^K$. The environment then reveals the context s_t to the learner, and the learner then proposes a probability distribution $p_t \in \Delta([K])$ over all arms. Finally the learner samples an action $a_t \in [K]$ according to p_t and receives the cost and risk associated to the chosen action: $c_t[a_t]$ and $r_t[a_t]$ (we denote $c[i]$ as the i'th element of vector c). The learner ideally wants to make a sequence of decisions that has low accumulative cost and also satisfies the constraint that related to the risk: $p_t^T r_t \leq \beta$ where $\beta \in [0,1]$ is a pre-defined threshold.

We address this problem by leveraging experts' advice. Given the expert set Π that consists of N experts $\{\pi_i\}_{i=1}^N$, where each expert $\pi \in \Pi : \mathcal{S} \to \Delta([K])$, gives advice by mapping from the context s to a probability distribution p over arms. The learner then properly combines the experts' advice $\{\pi_i(s)\}_{i=1}^N$ (e.g., compute the average $\sum_{i=1}^N \pi_i(s)/N$) to generate a distribution over all arms. With risk constraints, distributions over policies in Π could be strictly more powerful than any policy in Π itself. We aim to compete against this more powerful set. Given any distribution $w \in \Delta(\Pi)$, the mixed policy resulting from w can be regarded as: sample policy i according to w and then sample an arm according to $\pi_i(s)$. Though we do not place any statistical assumptions (e.g., i.i.d) on the sequence of cost vectors $\{c_t\}$ and risk vectors $\{r_t\}$, we assume the policy set Π is rich enough to satisfy the following assumption:

Assumption 2.2. *The set of distributions from $\Delta(\Pi)$ whose mixed policies satisfy all risk constraints in expectation is non-empty:*

$$\mathcal{P} \doteq \{w \in \Delta(\Pi) : \mathbb{E}_{i\sim w, j\sim\pi_i(s_t)} r_t[j] \leq \beta, \forall t\} \neq \emptyset.$$

Namely we assume that the distribution set $\Delta(\Pi)$ is rich enough such that there always exists at least one mixed policy that can satisfy all risk constraints in hindsight. Similar to the full information setting, competing against the set of mixed policies that satisfy the constraint on average, namely $\mathcal{P}' = \{w \in \Delta(\Pi) : \sum_{t=1}^{T} \mathbb{E}_{i\sim w, j\sim\pi_i(s_t)} r_t[j]/T \leq \beta\}$, is impossible in the partial information setting. Hence we define the best mixed policy in hindsight as:

$$w^* = \arg\min_{w\in\mathcal{P}} \sum_{t=1}^{T} \mathbb{E}_{i\sim w, j\sim\pi_i(s_t)} c_t[j]. \tag{3}$$

Given a sequence of decisions $\{a_t\}_{t=1}^T$ generated from some algorithm, we define average regret and average constraint violation as:

$$R_c = \frac{1}{T}\big[\sum_{t=1}^{T} c_t[a_t] - \sum_{t=1}^{T} \mathbb{E}_{i\sim w^*, j\sim \pi_i(s_t)} c_t[j]\big],$$

$$R_r = \frac{1}{T}\big[\sum_{t=1}^{T} (r_t[a_t] - \beta)\big].$$

The goal is to either minimize R_c and R_r in high probability or minimize the expected version $\bar{R}_c \doteq \mathbb{E}[R_c]$ and $\bar{R}_r \doteq \mathbb{E}[R_r]$ where the expectation is over the randomness of the algorithms.

3. Online Learning with Sequential Constraints

The online learning with adversarial sequential constraints setting is similar to the one considered in (Mannor et al., 2009; Jenatton et al., 2016) except that they only have a pre-defined fixed global constraint. However we find that their algorithms and analysis are general enough to extend to the online learning with adversarial sequential constraints. In (Mannor et al., 2009; Jenatton et al., 2016), the algorithms introduce a Lagrangian dual parameter and perform online gradient descent on x and online gradient ascent on the dual parameter. Since in this work we are eventually interested in reducing the contextual bandit problem to the full information online learning setting, simply adopting the OGD-based approaches from (Mannor et al., 2009; Jenatton et al., 2016) will not give a near optimal regret bound. Hence, developing the corresponding Multiplicative Weight (MW) update procedure is essential for a successful reduction from adversarial contextual bandit to full information online learning setting.

3.1. Algorithm

We use the same saddle-point convex concave formation from (Mannor et al., 2009; Jenatton et al., 2016) to design a composite loss function as:

$$\mathcal{L}_t(x, \lambda) = \ell_t(x) + \lambda f_t(x) - \frac{\delta\mu}{2}\lambda^2, \tag{4}$$

where $\delta \in \mathcal{R}^+$. Alg. 1 leverages online mirror descent (OMD) for updating the x (Line 6 and Line 7) and online gradient ascent algorithm for updating λ (Line 8). Note that if we use the square norm $\|x\|^2$ as the regularization function $R(x)$ in Alg. 1, we reveal the gradient descent based update procedure from (Mahdavi et al., 2012).

Algorithm 1 OCP with Sequential Constraints via OMD

```
1: Input: Learning rate $\mu$, mirror map $R$, parameter $\delta$ (used in $\mathcal{L}_t$).
2: Initialize $x_1 \in \mathcal{X}$ and $\lambda_1 = 0$.
3: for t = 1 to T do
4:     Learner proposes $x_t$.
5:     Receive loss function $\ell_t$ and constraint $f_t$.
6:     Set $\tilde{x}_{t+1}$ such that $\nabla R(\tilde{x}_{t+1}) = \nabla R(x_t) - \mu\nabla_x \mathcal{L}_t(x_t, \lambda_t)$.
7:     Projection: $x_{t+1} = \arg\min_{x\in\mathcal{X}} D_R(x, \tilde{x}_{t+1})$.
8:     Update $\lambda_{t+1} = \max\{0, \lambda_t + \mu\nabla_\lambda \mathcal{L}_t(x_t, \lambda_t))\}$.
9: end for
```

3.2. Analysis of Alg. 1

Throughout our analysis, we assume the regularization function $R(x)$ is α-strongly convex. For simplicity, we assume the number of rounds T is given and we consider the asymptotic property of Alg. 1 when T is large enough.

The algorithm should be really understood as running two no-regret procedures: (1) Online Mirror Descent on the sequence of loss $\{\mathcal{L}(x, \lambda_t)\}_t$ with respect to x and (2) Online Gradient ascent on the sequence of loss $\{\mathcal{L}(x_t, \lambda)\}_t$ with respect to λ. Instead of digging into the details of Online Mirror Descent and Online Gradient ascent, our analysis simply leverages the existing analysis of online mirror descent and online gradient ascent and show how to combine them to derive the regret bound and constraint violation bound for Alg. 1.

Theorem 3.1. *Let $R(\cdot)$ be a α-strongly convex function. Set $\mu = \sqrt{\frac{B}{T(D^2+G^2/\alpha)}}$ and $\delta = \frac{2G^2}{\alpha}$. For any convex loss $\ell_t(x)$, convex constraint $f_t(x) \leq 0$, under the assumption that $\mathcal{O} \neq \emptyset$, the family of algorithms induced by Alg. 1 have the following property:*

$$R_\ell \leq O(1/\sqrt{T}), \quad R_f \leq O(T^{-1/4}).$$

Proof Sketch of Theorem 3.1. Since the algorithm runs online mirror descent on the sequence of loss $\{\mathcal{L}_t(x, \lambda_t)\}_t$ with respect to x, using the existing results of online mirror descent (e.g., Theorem 4.2 and Eq. 4.10 from (Bubeck, 2015)), we know that for the computed sequence $\{x_t\}_t$:

$$\sum_{t=1}^{T}(\mathcal{L}_t(x_t, \lambda_t) - \mathcal{L}_t(x, \lambda_t))$$
$$\leq \frac{D_R(x, x_1)}{\mu} + \frac{\mu}{2\alpha}\sum_{t=1}^{T}\|\nabla_x \mathcal{L}_t(x_t, \lambda_t)\|_*^2, \tag{5}$$

for any $x \in \mathcal{X}$. Also, we know that the algorithm runs online gradient ascent on the sequence of loss $\{\mathcal{L}_t(x_t, \lambda)\}_t$ with respect to λ, using the existing analysis of online gradient descent (Zinkevich, 2003), we have for the computed sequence of $\{\lambda_t\}_t$:

$$\begin{aligned}&\sum_{t=1}^{T}\mathcal{L}_t(x_t,\lambda)-\sum_{t=1}^{T}\mathcal{L}_t(x_t,\lambda_t)\\&\leq\frac{1}{\mu}\lambda^2+\frac{\mu}{2}\sum_{t=1}^{T}(\frac{\partial\mathcal{L}_t(w_t,\lambda_t)}{\partial\lambda_t})^2,\end{aligned}\tag{6}$$

for any $\lambda\geq 0$.

Note that for $(\partial\mathcal{L}_t(x_t,\lambda_t)/\partial\lambda_t)^2=(f_t(x_t)-\delta\mu\lambda_t)^2\leq 2f_t^2(x_t)+2\delta^2\mu^2\lambda_t^2\leq 2D^2+\delta^2\mu^2\lambda_t^2$. Similarly for $\|\nabla_x\mathcal{L}_t(x_t,\lambda_t)\|_*^2$, we also have:

$$\begin{aligned}&\|\nabla_x\mathcal{L}_t(x_t,\lambda_t)\|_*^2\leq 2\|\nabla\ell_t(x_t)\|_*^2+2\|\lambda_t\nabla f_t(x_t)\|_*^2\\&\leq 2G^2(1+\lambda_t^2),\end{aligned}\tag{7}$$

where we first used triangle inequality for $\|\nabla_x\mathcal{L}_t(x_t,\lambda_t)\|_*$ and then use the inequality of $2ab\leq a^2+b^2,\forall a,b\in\mathcal{R}^+$. Note that we also assumed that the norm of the gradients are bounded as $max(\|\nabla\ell_t(x_t)\|_*,\|\nabla f_t(x_t)\|_*)\leq G\in\mathcal{R}^+$. Now sum Inequality 5 and 6 together, we get:

$$\begin{aligned}&\sum_t\mathcal{L}_t(x_t,\lambda)-\mathcal{L}_t(x,\lambda_t)\\&\leq\frac{2D_R(x,x_1)+\lambda^2}{2\mu}+\sum_t\mu(D^2+\delta^2\mu^2\lambda_t^2)\\&\quad+\sum_t\frac{\mu G^2}{\alpha}(1+\lambda_t^2)\\&=\frac{2D_R(x,x_1)+\lambda^2}{2\mu}+T\mu(D^2+\frac{G^2}{\alpha})\\&\quad+\mu(\delta^2\mu^2+\frac{G^2}{\alpha})\sum\lambda_t^2.\end{aligned}\tag{8}$$

Substitute the form of $\mathcal{L}_t$ into the above inequality, we have:

$$\begin{aligned}&\sum_t(\ell_t(x_t)-\ell_t(x))+\sum_t(\lambda f_t(x_t)-\lambda_t f_t(x))\\&\quad+\frac{\delta\mu}{2}\sum_t\lambda_t^2-\frac{\delta\mu T}{2}\lambda^2\leq\frac{2D_R(x,x_1)+\lambda^2}{2\mu}\\&\quad+T\mu(D^2+\frac{G^2}{\alpha})+\mu(\delta^2\mu^2+\frac{G^2}{\alpha})\sum_t\lambda_t^2.\end{aligned}\tag{9}$$

Note that from our setting of μ and δ we can verify that $\delta\geq\delta^2\mu^2+G^2/\alpha$, we can remove the term $\sum_t\lambda_t^2$ in the above inequality.

Without the term $\sum_t\lambda_t^2$, to upper bound the regret on loss ℓ_t, let us set $\lambda=0$ and $x=x^*$, we get:

$$\begin{aligned}&\sum_t(\ell_t(x_t)-\ell_t(x^*))\leq\frac{2D_R(x,x_1)}{2\mu}+T\mu(D^2+G^2/\alpha)\\&\leq 2\sqrt{D_R(x,x_1)T(D^2+G^2/\alpha)}=O(\sqrt{T}),\end{aligned}$$

with $\mu=\sqrt{D_R(x,x_1)/(T(D^2+G^2/\alpha))}$. To upper bound $\sum_t f_t(x_t)$, we first observe that we can lower bound $\sum_{t=1}^{T}\ell_t(x_t)-\min_x\sum_{t=1}^{T}\ell_t(x)\geq -2FT$, where F is the upper bound of $\ell(\cdot)$. Replace $\sum_t\ell_t(x_t)-\ell_t(x)$ by $-2FT$ in Eq. 9, and set $\lambda=(\sum_t f_t(x_t))/(\delta\mu T+1/\mu)$ (here we assume $\sum_t f_t(x_t)\geq 0$, otherwise we prove the theorem), we can show that:

$$\begin{aligned}&(\sum_{t=1}^{T}f_t(x_t))^2\leq\frac{8G^2}{\alpha}D_R(x,x_1)+2(D^2+\frac{G^2}{\alpha})T\\&\quad+T^{3/2}\sqrt{8F^2G^2/\alpha}\end{aligned}\tag{10}$$

The RHS of the above inequality is dominated by the term $T^{3/2}\sqrt{8F^2G^2/\alpha}$ when T approaches to infinity. Hence, we get $\sum_{t=1}^{T}f_t(x_t)=O(T^{3/4})$. □

As we can see that if we replace $R(x)$ with $\|x\|_2^2$ in Alg. 1, we reveal a gradient descent based update procedure that is almost identical to the one in (Mahdavi et al., 2012). When x is restricted to a simplex, to derive the multiplicative weight update procedure, we replace $R(x)$ with the negative entropy regularization $\sum_i x[i]\ln(x[i])$ and we can achieve the following update steps for x:

$$x_{t+1}[i]=\frac{x_t[i]\exp(-\mu\nabla_x\mathcal{L}_t(x_t,\lambda_t)[i])}{\sum_{j=1}^{d}x_t[j]\exp(-\mu\nabla_x\mathcal{L}_t(x_t,\lambda_t)[j])}.$$

We refer readers to (Shalev-Shwartz, 2011; Bubeck, 2015) for the derivation of the above equation.

4. Contextual Bandits With Risk Constraints

When contexts, costs and risks are i.i.d sampled from some unknown distribution, then our problem setting can be regarded as a special case of the setting of contextual bandit with global objective and constraint (CBwRC) considered in (Agrawal et al., 2015). In (Agrawal et al., 2015), the algorithm also leverages Lagrangian dual variable. The difference is that in i.i.d setting the dual parameter is fixed with respect to the underlying distribution and hence it is possible to estimate the dual variable. For instance one can uniformly pull arms with a fixed number of rounds at the beginning to gather information for estimating the dual variable and then use the estimated dual variable for all remaining rounds. However in the adversarial setting, this nice trick will fail since the costs and risks are possibly sampled from a changing distribution. We have to rely on OCP algorithms to keep updating the dual variable to adapt to adversarial risks and costs.

4.1. Algorithm

Our algorithm EXP4.R (EXP4 with Risk constraints) (Alg. 2) extends the EXP4 (Auer et al., 2002) algorithm

For simplicity we assumed T is large enough to be larger than any given constant.

Algorithm 2 EXP4 with Risk Constraints (EXP4.R)

```
1:  Input: Policy set $\Pi$.
2:  Initialize $w_1 = [1/N, ..., 1/N]^T$ and $\lambda_1 = 0$.
3:  for t = 1 to T do
4:      Receive context $s_t$.
5:      Query experts to get advice $\pi_i(s_t), \forall i \in [N]$.
6:      Set $p_t = \sum_{i=1}^{N} w_t[i]\pi_i(s_t)$.
7:      Draw action $a_t$ randomly from distribution $p_t$.
8:      Receive cost $c_t[a_t]$ and risk $r_t[a_t]$.
9:      Set the cost vector $\hat{c}_t \in R^K$ and the risk vector $\hat{r}_t \in R^K$ as follows: for all $i \in [K]$
            $\hat{c}_t[i] = \frac{c_t[i]\mathbb{1}(a_t = i)}{p_t[i]}, \quad \hat{r}_t[i] = \frac{r_t[i]\mathbb{1}(a_t = i)}{p_t[i]}.$
10:     For each expert $j \in [N]$, set:
            $\hat{y}_t[j] = \pi_j(s_t)^T \hat{c}_t, \quad \hat{z}_t[j] = \pi_j(s_t)^T \hat{r}_t.$
11:     Compute $w_{t+1}$, for $i \in [|\Pi|]$:
            $w_{t+1}[i] = \frac{w_t[i]\exp\big(-\mu(\hat{y}_t[i] + \lambda_t \hat{z}_t[i])\big)}{\sum_{j=1}^{|\Pi|} w_t[j]\exp\big(-\mu(\hat{y}_t[j] + \lambda_t \hat{z}_t[j])\big)}.$
12:     Compute $\lambda_{t+1}$:
            $\lambda_{t+1} = \max\{0, \lambda_t + \mu(w_t^T \hat{z}_t - \beta - \delta\mu\lambda_t)\}.$
13: end for
```

to carefully incorporating the risk constraints for updating the probability distribution w over all policies. At each round, it first uses the common trick of importance weighting to form unbiased estimations of cost vector $\hat{c}$ and risk vector $\hat{r}$. Then the algorithm uses the unbiased estimations of cost vector and risk vector to form unbiased estimations of the cost $\hat{y}[i]$ and risk $\hat{z}[i]$ for each expert i. EXP4.R then starts behaving differently than EXP4. EXP4.R introduces a dual variable λ and combine the cost and risk together as $\mathcal{L}_t(w, \lambda) = w^T\hat{y}_t + \lambda(w^T\hat{z}_t - \beta) - \frac{\delta\mu}{2}\lambda^2$. We then use Alg. 1 with the negative entropy regularization as a black box online learner to update the weight w and the dual variable λ.

The proposed algorithm EXP4.R in general is computationally inefficient since similar to EXP4 and EXP4.P, it needs to maintain a probability distribution over the policy set. Though there exist computationally efficient algorithms for stochastic contextual bandits and hybrid contextual bandits, we are not aware of any computationally efficient algorithm for adversarial contextual bandits, even without risk constraints.

4.2. Analysis of EXP4.R

We provide a reduction based analysis for EXP4.R by first reducing EXP4.R to Alg. 1 with negative entropic regularization. For the following analysis, let us define $y_t[j] = \pi_j(s_t)^T c_t$ and $z_t[j] = \pi_j(s_t)^T r_t$, which stand for the expected cost and risk for policy j at round t.

Let us define $\mathcal{L}_t(w, \lambda) = w^T\hat{y}_t + \lambda(w^T\hat{z}_t - \beta) - \frac{\delta\mu}{2}\lambda^2$. The multiplicative weight update in Line 11 can be regarded as running Weighted Majority on the sequence of loss $\{\mathcal{L}_t(w, \lambda_t)\}_t$, while the update rule for λ in Line 12 can be regarded as running Online Gradient Ascent on the sequence of loss $\{\mathcal{L}_t(w_t, \lambda)\}_t$. Directly applying the classic analysis of Weighted Majority (Shalev-Shwartz, 2011) on the generated sequence of weights $\{w_t\}_t$ and the classic analysis of OGD (Zinkevich, 2003) on the generated sequence of dual variables $\{\lambda_t\}_t$, we get the following lemma:

Lemma 4.1. *With the negative entropy $\sum_i x[i]\ln(x[i])$ as the regularization function for $R(x)$ in Alg. 1, running Alg. 1 on the sequence of linear loss functions $\ell_t(w) = w^T\hat{y}_t$ and linear constraint $f_t(w) = w^T\hat{z}_t - \beta \le 0$, we have:*

$$\sum_{t=1}^{T}\mathcal{L}_t(w_t, \lambda) - \sum_{t=1}^{T}\mathcal{L}_t(w, \lambda_t) \le \frac{\lambda^2}{\mu} + \frac{\ln(|\Pi|)}{\mu} + \frac{\mu}{2}\sum_{t=1}^{T}\Big(\big(\sum_{i=1}^{|\Pi|} w_t[i](2\hat{y}_t[i]^2 + 2\lambda_t^2\hat{z}_t[i]^2)\big) + (w_t^T\hat{z}_t - \beta - \delta\mu\lambda_t)^2\Big). \tag{11}$$

We defer the proof of the above lemma to Appendix. The EXP4.R algorithm has the following property:

Theorem 4.2. *Set $\mu = \sqrt{\ln(|\Pi|)/(T(K+4))}$ and $\delta = 3K$. Assume $\mathcal{P} \neq \emptyset$. EXP4.R has the following property:*

$$\bar{R}_c = O(\sqrt{K\ln(|\Pi|)/T}),$$
$$\bar{R}_r = O(T^{-1/4}(K\ln(|\Pi|))^{1/4}).$$

Proof Sketch of Theorem 4.2. The proof consists of a combination of the analysis of EXP4 and the analysis of Theorem 3.1. We defer the full proof in Appendix C. We first present several known facts. First, we have $w_t^T\hat{z}_t = r_t[a_t] \le 1$ and $w_t^T\hat{y}_t = c_t[a_t] \le 1$.

For $\mathbb{E}_{a_t\sim p_t}(w_t^T\hat{z}_t - \beta)^2$, we can show that: $\mathbb{E}_{a_t\sim p_t}(w_t^T\hat{z}_t - \beta)^2 \le 2 + 2\beta^2 \le 4$.

It is also straightforward to show that $\mathbb{E}_{a_t\sim p_t}\hat{y}_t = y_t$ and $\mathbb{E}_{a_t\sim p_t}\hat{z}_t = z_t$. It is also true that $\mathbb{E}_{a_t\sim p_t}\sum_{i=1}^{|\Pi|} w_t[i]\hat{y}_t[i]^2 \le K$ and $\mathbb{E}_{a_t\sim p_t}\sum_{i=1}^{|\Pi|} w_t[i]\hat{z}_t[i]^2 \le K$.

Now take expectation with respect to the sequence of decisions $\{a_t\}_t$ on LHS of Inequality 11:

$$\mathbb{E}_{\{a_t\}_t}\sum_{t=1}^{T}\Big[\mathcal{L}_t(w_t,\lambda)-\mathcal{L}_t(w,\lambda_t)\Big]$$

$$=\sum_{t=1}^{T}\big[\mathbb{E}c_t[a_t]+\lambda(\mathbb{E}r_t[a_t]-\beta)-y_t^T w-\lambda_t(z_t^T w-\beta)$$

$$+\frac{\delta\mu}{2}\lambda_t^2\big]-\frac{\delta\mu T}{2}\lambda^2 \tag{12}$$

Now take the expectation with respect to $a_1,...,a_T$ on the RHS of inequality 11, we can get:

$$\mathbb{E}[\textit{RHS of Inequality 11}]\leq\frac{\lambda^2}{\mu}+\frac{\ln|\Pi|}{\mu}$$

$$+\mu T(K+4)+\mu(K+\delta^2\mu^2)\sum_{t=1}^{T}\lambda_t^2. \tag{13}$$

Now we can chain Eq. 12 and 13 together and use the same technique that we used in the analysis of Theorem 3.1. Chain Eq. 12 and 13 together and set w to w^* and $\lambda=0$, it is not hard to show that:

$$\mathbb{E}\big(\sum_{t=1}^{T}c_t[a_t]-\sum_{t=1}^{T}y_t^T w^*\big)$$

$$\leq 2\sqrt{\ln(|\Pi|)T(K+4)}=O(\sqrt{TK\ln(|\Pi|)}),$$

where $\mu=\sqrt{\ln(|\Pi|)/(T(K+4))}$. Set $\lambda=(\sum_t(\mathbb{E}r_t[a_t]-\beta))/(\delta\mu T+2/\mu)$, we can get:

$$(\sum_{t=1}^{T}(\mathbb{E}r_t[a_t]-\beta))^2\leq(2\delta\mu T+4/\mu)\Big(2T$$

$$+2\sqrt{\ln(|\Pi|)T(K+4)}\Big) \tag{14}$$

Substitute $\mu=\sqrt{\ln(|\Pi|)/(T(K+4))}$ back to the above equation, it is easy to verity that:

$$(\sum_{t=1}^{T}(\mathbb{E}r_t[a_t]-\beta))^2\leq O\big(T^{3/2}(K\ln(|\Pi|))^{1/2}\big). \tag{15}$$

□

4.3. Extension To High-Probability Bounds

The regret bound and constraint violation bound of EXP4.R hold in expectation. In this section, we present an algorithm named EXP4.P.R, which achieves high-probability regret bound and constraint violation bound. The algorithm EXP4.P.R, as indicated by its name, is built on the well-known EXP4.P algorithm. In this section for the convenience of analysis, without loss of generality, we are going to assume that for any cost vector c and risk vector r, we have $c[i]\in[-1,0]$, $r[i]\leq[-1,0],\forall i\in[K]$, and $\beta\in[-1,0]$.

The whole framework of the algorithm is similar to the one of EXP4.R, with *only one* modification. For notation simplicity, let us define $\tilde{x}_t[i]=\hat{y}_t[i]+\lambda_t\hat{z}_t[i]$. Note that $\tilde{x}_t[i]$ is an unbiased estimate of $y_t[i]+\lambda_t z_t[i]$. EXP4.P.R modifies EXP4.R by replacing the update procedure for w_{t+1} in Line 11 in Alg. 2 with the following update step:

$$w_{t+1}[i]=\frac{w_t[i]\exp(-\mu(\tilde{x}_t[i]-\kappa\sum_{k=1}^{K}\frac{\pi_i(s_t)[k]}{p_t[k]}))}{\sum_{j=1}^{|\Pi|}w_t[j]\exp(-\mu(\tilde{x}_t[j]-\kappa\sum_{k=1}^{K}\frac{\pi_j(s_t)[k]}{p_t[k]}))},$$

where κ is a constant that will be defined in the analysis of EXP4.P.R. We refer readers to Appendix D for the full version of EXP4.P.R. Essentially, similar to EXP3.P and EXP4.P, we add an extra term $-\kappa\sum_{k=1}^{K}\frac{\pi_j(s_t)[k]}{p_t[k]}$ to $\tilde{x}_t[i]$. Though $\tilde{x}_t[i]-\kappa\sum_{k=1}^{K}\frac{\pi_j(s_t)[k]}{p_t[k]}$ is not an unbiased estimation of $y_t[i]+\lambda_t z_t[i]$ anymore, as shown in Lemma D.3 in Appendix D.2, it enables us to upper bound $\sum_t\tilde{x}_t[i]-\kappa\sum_{k=1}^{K}\frac{\pi_j(s_t)[k]}{p_t[k]}$ using $\sum_t y_t[i]+\lambda_t z_t[i]$ with high probability.

We show that EXP4.P.R has the following performance guarantees:

Theorem 4.3. *Assume $\mathcal{P}\neq\emptyset$. For any $\epsilon\in(0,1/2)$, $\nu\in(0,1)$, set $\mu=\sqrt{\frac{\ln(|\Pi|)}{(3K+4)T}}$, $\kappa=\sqrt{\frac{(1+T^{\epsilon})\ln(|\Pi|/\nu)}{TK}}$, and $\delta=T^{-\epsilon+1/2}K$, we have that with probability at least $1-\nu$:*

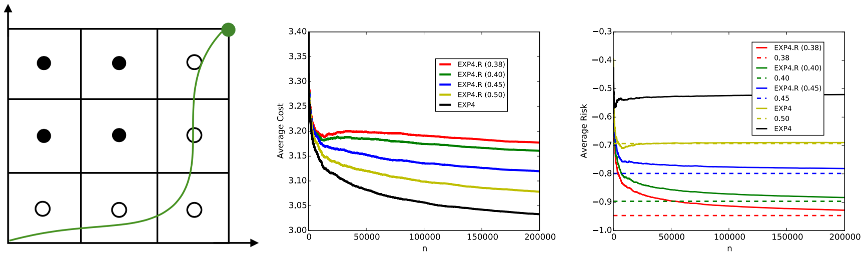
$$R_c=O(\sqrt{T^{\epsilon-1}K\ln(|\Pi|/\nu)}),$$

$$R_r=O(T^{-\epsilon/2}\sqrt{K\ln(|\Pi|)}). \tag{16}$$

The above theorem introduces a trade-off between the regret of cost and the constraint violation. As $\epsilon\to 0$, we can see that the regret of cost approaches to the near-optimal one $\sqrt{TK\ln(|\Pi|)}$, but the average risk constrain violation approaches to a constant. Based on specific applications, one may set a specific $\epsilon\in(0,0.5)$ to balance the regret and the constraint violation. For instance, for $\epsilon=1/3$, one can show that the cumulative regret is $O(T^{2/3}\sqrt{K\ln(|\Pi|)})$ and the average constraint violation is $\tilde{O}(T^{-1/6})$. Note that if one simply runs EXP4.R proposed in the previous section, it is impossible to achieve the regret rate $O(T^{2/3}\sqrt{\ln(|\Pi|)})$ in a high probability statement. As shown in (Auer et al., 2002), for EXP4 the cumulative regret on the order of $O(T^{3/4})$ was possible.

The difficulty of achieving a high probability statement with optimal cumulative regret $O(\sqrt{TK\ln(|\Pi|)})$ and cumulative constraint violation rate $\tilde{O}(T^{3/4})$ is from the Lagrangian dual variable λ. The variance of $\hat{z}_t$ is proportional to $1/p_t[a_t]$. With λ_t, the variance of $\lambda_t\hat{z}_t$ scales as

EXP4.R becomes the same as EXP4 when we set all risks and β to zeros.



(a) Environment

(b) Cost

(c) Risk

Figure 1. (a) Environment set up and a safe trajectory from the initial position to the goal region (green). (b,c) The performance of EXP4.R under different values of risk thresholds (y-axis is in log scale).

$\lambda_t^2/p_t[a_t]$. As we show in Lemma D.2 in Appendix D.2, λ_t could be as large as $|\beta|/(\delta\mu)$. Depending on the value of δ, μ, λ_t could be large, e.g., $\Theta(\sqrt{T})$ if δ is a constant and $\mu = \Theta(1/\sqrt{T})$. Hence compared to EXP4.P, the Lagrangian dual variable in EXP4.P.R makes it more difficult to control the variance of $\tilde{x}_t$, which is an unbiased estimation of $y_t[i] + \lambda_t z_t[i]$. This is exactly where the trade-off ϵ comes from: we can tune the magnitude of δ to control the variance of $\tilde{x}_t$ and further control the trade-off between regret and risk violation . How to achieve total regret $O(\sqrt{TK\ln(|\Pi|)})$ and cumulative constraint violation $O(T^{3/4})$ in high probability is still an open problem.

5. Simulation

We test our algorithms on a simple synthetic robotics reactive control task. The 2-D environment, shown in Fig. 1a is set up as follows. We divide the environment into 9 cells and each cell is associated by a waypoint (black dot or black circle). The black dots are associated with risk 1 and stands for dangerous areas while the circle dots are associated with risk zeros (i.e., safe regions). The state of the agent is its 2-D position and for each state, we compute the RBF feature s with respect 9 waypoints and then normalize the feature s such that $\sum_{i=1}^{9} s[i] = 1$. The risk for feature s is the inner product between s and the 9-dimension risk vector. The reward of action a is $\max(0, d_1 - d_2)$ (we simply negate the reward to fake a cost to feed to our algorithms), where d_1 stands for the old distance to goal and d_2 is the new distance to goal after taken action a (hence a non-linear reward mapping). We designed 4 actions for the agent: it can move up, right, left, and down with a fixed small constant distance. We design 4^9 experts where each expert π_i is a 9 dimensional vector, where the element in each dimension belongs to the action set (Left,Right, Up, Down) (namely each expert π_i suggests an action for every waypoint). Given a feature s, the suggestion $\pi_i(s)$ is computed as $\pi_i(s)[i] = \sum_{k,\pi_i[k]=i} s[k]$. Note that this is similar to the setting consider in (Beygelzimer et al., 2011). Note that the class of experts can be represented by a depth-9 tree with 4 branches. Hence computational efficient weighted majority algorithm implementation could be used here (Cesa-Bianchi & Lugosi, 2006). But in this work, we temporarily lighten computational burden by paralleling computing. We initialize the weight over all experts uniformly and let the EXP4.R algorithm picks decision for the agent. The agent executes the decision, receives risk and cost. The agent is reset to the initial position once it reaches the goal region or outside of the map too much.

Fig. 1b and 1c show the performance of EXP4.R under different values of risk threshold β. We observe that EXP4.R can ensure that the average risk converges to the threshold β, while keep decreasing the average cost simultaneously. A lower risk threshold usually results a slower decrease in the average cost, which is consistent to the theorems since a smaller risk threshold leads to a smaller $\mathcal{P}$. Compare to the performance of EXP4, we see the EXP4 can offer faster decrease rate for cost (as it can compete against the whole policy class set), but it cannot control risk, e.g, it discovers paths that cut through the middle high-risk cell.

6. Conclusion

We study safe sequential decision making problem under the format of adversarial contextual bandits with adversarial sequential risk constraints. We provide safety-aware algorithms that can satisfy the long-term risk constraint while achieve near-optimal regret in terms of minimizing costs. The proposed two algorithm, EXP4.R and EXP4.P.R, are built on the existing EXP4 and EXP4.P algorithms: EXP4.R achieves near-optimal regret and satisfies the long-term constraint in expectation while EXP4.P.R achieves similar theoretical bounds with high probability, with a tradeoff that can trade the constraint violation for regret and vice versa. Same as EXP4 and EXP4.P, the computational complexity of a simple implementation of our algorithms per step is linear with respect to the size of the expert class. However for expert class that has special structures such as trees (as we used in our simulation) or graphs, one can usually design efficient implementation. We leave the efficient implementation for real robotics control as a future work.

Acknowledgement

The research work was done during Wen Sun's internship at Microsoft Research, Redmond.

References

Agarwal, Alekh, Hsu, Daniel, Kale, Satyen, Langford, John, Li, Lihong, and Schapire, Robert. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of The 31st International Conference on Machine Learning*, pp. 1638–1646, 2014.

Agrawal, Shipra and Devanur, Nikhil R. Linear contextual bandits with knapsacks. *arXiv preprint arXiv:1507.06738*, 2015.

Agrawal, Shipra, Devanur, Nikhil R, and Li, Lihong. Contextual bandits with global constraints and objective. *arXiv preprint arXiv:1506.03374*, 2015.

Amodei, Dario, Olah, Chris, Steinhardt, Jacob, Christiano, Paul, Schulman, John, and Mané, Dan. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Auer, Peter, Cesa-Bianchi, Nicolo, Freund, Yoav, and Schapire, Robert E. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Badanidiyuru, Ashwinkumar, Kleinberg, Robert, and Slivkins, Aleksandrs. Bandits with knapsacks. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pp. 207–216. IEEE, 2013.

Badanidiyuru, Ashwinkumar, Langford, John, and Slivkins, Aleksandrs. Resourceful contextual bandits. In *COLT*, pp. 1109–1134, 2014.

Beck, Amir and Teboulle, Marc. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Beygelzimer, Alina, Langford, John, Li, Lihong, Reyzin, Lev, and Schapire, Robert E. Contextual bandit algorithms with supervised learning guarantees. In *AISTATS*, pp. 19–26, 2011.

Bubeck, Sébastien. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8 (3-4):231–357, 2015.

Bubeck, Sébastien, Cesa-Bianchi, Nicolò, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Cesa-Bianchi, Nicolo and Lugosi, Gábor. *Prediction, learning, and games*. Cambridge university press, 2006.

Ding, Wenkui, Qin, Tao, Zhang, Xu-Dong, and Liu, Tie-Yan. Multi-armed bandit with budget constraint and variable costs. In *AAAI*, 2013.

Jenatton, Rodolphe, Huang, Jim, and Archambeau, Cédric. Adaptive algorithms for online convex optimization with long-term constraints. *ICML*, 2016.

Langford, John and Zhang, Tong. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pp. 817–824, 2008.

Madani, Omid, Lizotte, Daniel J, and Greiner, Russell. The budgeted multi-armed bandit problem. In *International Conference on Computational Learning Theory*, pp. 643–645. Springer, 2004.

Mahdavi, Mehrdad, Jin, Rong, and Yang, Tianbao. Trading regret for efficiency: online convex optimization with long term constraints. *The Journal of Machine Learning Research*, 13(1): 2503–2528, 2012.

Mannor, Shie, Tsitsiklis, John N, and Yu, Jia Yuan. Online learning with sample path constraints. *The Journal of Machine Learning Research*, 10:569–590, 2009.

Shalev-Shwartz, Shai. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning*, 4(2): 107–194, 2011.

Zinkevich, Martin. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *International Conference on Machine Learning (ICML 2003)*, pp. 421–422, 2003.