Safe Optimal Design with Applications in Off-Policy Learning

Ruihao Zhu
Purdue University, Krannert School of Management

Branislav Kveton
Amazon*

Abstract

Motivated by practical needs in online experimentation and off-policy learning, we study the problem of *safe optimal design*, where we develop a data *logging policy* that efficiently explores while achieving competitive rewards with a baseline *production policy*. We first show, perhaps surprisingly, that a common practice of mixing the production policy with uniform exploration, despite being safe, is sub-optimal in maximizing information gain. Then we propose a safe optimal logging policy for the case when no side information about the actions' expected rewards is available. We improve upon this design by considering side information and also extend both approaches to a large number of actions with a linear reward model. We analyze how our data logging policies impact errors in off-policy learning. Finally, we empirically validate the benefit of our designs by conducting extensive experiments.

1 INTRODUCTION

Experimentation is used widely to test the effectiveness of new actions and develop policies that efficiently allocate traffic to different actions. For instance, online platforms constantly conduct large-scale experiments for market mechanism design, webpage layout, and product recommendation. With ever-increasing demand for experiments, several companies have developed infrastructure to carry them out at scale (see, *e.g.*, Optimizely (2021a); Google Optimize (2021)). Two popular types of experimentation techniques are *non-adaptive* A/B testing (*e.g.*, randomized controlled experiments (Gallo, 2017; Optimizely, 2021b)) for fixed policies and *adaptive* online learning (*e.g.*, multi-armed bandit (Auer et al., 2002; Even-Dar et al., 2002)) for a dynamically updated policy. The policies in A/B testing are typically learned *offline* from data collected by some exploratory data logging policy. In contrast, online learning allocates more traffic to better-performing actions in a real time fashion.

At a first glance, online learning seems to be more efficient than A/B testing due to lower experimentation cost (Li et al., 2010; Schwartz et al., 2017). Nevertheless, despite a major progress in designing near-optimal online learning algorithms over the past decades (Valko et al., 2013; Jamieson and Nowak, 2014; Chen et al., 2014; Soare et al., 2014; Abernethy et al., 2016; Qin et al., 2017; Besbes et al., 2018; Cheung et al., 2019), many challenges hinder their wide adoption in practice:

- **Infrastructure:** To implement online learning algorithms in real world, it is necessary to collect responses and update traffic allocation in near real time, which poses significant challenges to the computational infrastructure (Chen et al., 2020; Simchi-Levi and Xu, 2021; Ruan et al., 2021).
- **Logged-Data Estimation Error and Bias:** Due to the reward maximizing nature, online learning algorithms allocate less traffic to actions with poor historical performance. Therefore, it is common to encounter a major estimation error when estimating their expected rewards from the logged data. In many applications (*e.g.*, ads design and webpage layout), it is important to understand the performance of such actions (Danilchik, 2020). Even worse, existing works (see, *e.g.*, Nie et al. (2017); Shin et al. (2019)) showed that a direct application of maximum likelihood estimation to adaptively collected data can result in a significant bias. The debiasing is challenging because the logging policy is adapted over time to the collected data.
- **Excessive Initial Exploration:** Non-Bayesian bandit algorithms tend to explore extensively in the

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

*The work started while being at Google Research.

initial rounds. This can have a major impact on user experience and lead to early termination of the experiment in online marketplaces and clinical trials (Wu et al., 2016; Bastani et al., 2021a), for instance.

For these reasons, the most common practice in the industry is to learn policies offline and A/B test them before they are deployed. The process of learning the policies offline is known as *off-policy evaluation and optimization* (Li et al., 2011; Dudık et al., 2014; Swaminathan and Joachims, 2015).

Off-policy evaluation and optimization crucially rely on sufficiently explored logged data to drawn conclusions about alternative policies. When the data are collected, it is typically necessary to satisfy *safety constraints*, which prohibit too costly experimentation. To strike the balance, a common practice in the industry is to mix a baseline *production policy* with randomized actions. This results in a logging policy that explores, as it allocates traffic to all actions; but is also safe, since the production policy is followed frequently. As an example, if the logging policy has to perform as well as 95% of the production policy, then 95% of the traffic is allocated to the production policy, while the rest is randomly allocated to all actions.

After the logged data are collected, they are used to estimate the performance of candidate policies (Li et al., 2011; Dudık et al., 2014; Swaminathan and Joachims, 2015). Whether the logging policy is statistically efficient and suitable for the goal is rarely questioned. Ironically, the estimation errors in off-policy learning critically depend on the quality of the logged data. This raises an important question of *how to design a logging policy that is both safe and collects high-quality data*. In this work, we study this question through the lens of the *G-optimal design* (*i.e.*, globally-optimal design (Kiefer and Wolfowitz, 1960)). In the G-optimal design, the goal is to design a data logging policy that minimizes (a proxy of) the maximal variance in estimating the actions' expected rewards. Motivated by practical safety considerations in experimentation (Wu et al., 2016), we instantiate the safety constraint as follows: the expected reward of the logging policy is at least an α fraction of that of the production policy.

We make the following contributions: We first show, perhaps surprisingly, that following the production policy for an α fraction of the time and uniformly exploring otherwise is sub-optimal. Next we propose a water-filling algorithm that solves our problem optimally when no side information about the actions' expected rewards is available. We improve upon this design by considering side information, and also extend both approaches to a large number of actions with a linear reward model. We apply our results to off-policy evaluation and optimization, and demonstrate how our logging policy can provide performance guarantees for this setting. Finally, we conduct extensive numerical experiments to demonstrate the performance of our approaches.

1.1 Additional Related Works

Safe exploration has been a topic of many papers, some of which we review below. In summary, all prior formulations of this problem differ from our work and are not directly comparable.

Wu et al. (2016) proposed a bandit algorithm that conservatively improves upon a default action. The key idea is to take the default action α fraction of time and improve it over time, with provably better actions with a high probability. This work was generalized to linear bandits by Kazerouni et al. (2017) and to combinatorial action spaces, such as in online learning to rank, by Li et al. (2019). Our work is similar to these works only by a similar safety constraint. We learn the most exploratory policy under a safety constraint that collects useful data for future off-policy estimation and optimization, rather than continuously improve an online policy.

Another related problem is off-policy optimization with a safety constraint, where the learned policy improves over a logging policy with a high probability (Thomas et al., 2015; Laroche et al., 2019). In a sense, these works solve an opposite problem to ours. They learn policies with enough support to improve over the logging policy, while we explore to improve the support for future off-policy estimation and optimization.

G-optimal design and its variants have also been applied to regret minimization and best-arm identification in multi-armed bandit (Audibert et al., 2010; Bubeck et al., 2010; Karnin et al., 2013; Azizi et al., 2021; Yang and Tan, 2021), but safety was not considered in these works.

2 PROBLEM FORMULATION

Notations: Let $\mathcal{A} = [K] := \{1, \dots, K\}$ be a tabular *action set*. When action $a \in \mathcal{A}$ is taken, we observe its stochastic reward with (initially) unknown mean $\bar{r}(a) \in [0, 1]$. A *policy* $\pi : \mathcal{A} \to [0, 1]$ is a probability distribution on $\mathcal{A}$ and we denote by Π the set of all possible policies. To simplify notation, we use $\bar{r}$ and π to denote the vectorized expected rewards and the policy, *i.e.*, $\bar{r} = (\bar{r}(1), \dots, \bar{r}(K))^\top$ and $\pi = (\pi(1), \dots, \pi(K))^\top$. The expected reward of policy π is thus $V(\pi) = \pi^\top \bar{r}$. For any $p \geq 0$, we define $\|\cdot\|_p$ as the p-norm and $\Delta_{k-1} = \{x \in \mathbb{R}^k : x \geq 0, \|x\|_1 \leq 1\}$ as the k-dimensional simplex. For any $c \in \mathbb{R}$, we use $\boldsymbol{c}_k$ to

denote the k-dimensional vector with all entries equal to c.

Tabular Safe Optimal Design Setup: To overcome the challenges posed by online learning-based experimentation (see Section 1), we deploy a static *exploratory data logging policy* π_e for a certain time interval to carry out experimentation. For each time step of this interval, we randomly select an action according to π_e and observe the corresponding realized random reward. We seek to leverage the collected data to estimate each action's expected reward $\bar{r}(\cdot)$ and to further compute the near-optimal policy $\pi_* = \arg\max_{\pi\in\Pi} V(\pi)$ offline (see Section 3.3 for more details). Before formally introducing our objective, we first describe our criteria in developing π_e :

- **Information Gain:** The quality of our logging policy π_e is measured by $g(\pi) = \max_{a\in\mathcal{A}} 1/\pi(a)$. Intuitively, $g(\pi)$ is proportional to the maximum width of a *high-probability confidence interval* over $a \in \mathcal{A}$ (see *e.g.*, Section 21.1 of Lattimore and Szepesvari (2018)). Thus it measures how well we can estimate the unknown expected rewards and compute a near-optimal policy. A sensible objective is to find π_e that minimizes $g(\pi_e)$. Note that $g(\pi)$ is a special case of the G-optimal design objective (Kiefer and Wolfowitz, 1960) and without any constraint, we can set $\pi_e(a) = 1/K$ for all $a \in \mathcal{A}$ to maximize information gain.
- **Safety:** To avoid a potentially high cost in deploying π_e, we demand that π_e's expected reward is at least $\alpha \in [0, 1]$ of that of a *baseline production policy* π_0 for any instance of expected rewards $\bar{r}$. Specifically, $V(\pi_e) \geq \alpha V(\pi_0)$, where $\alpha \in [0, 1]$ is a *safety parameter*. We remark that the safety constraint could be defined alternatively as that π_e's expected reward is at most α less than that of π_0's. Our forthcoming results would apply to this case as well.

Objective: Formally, we want to design π_e that simultaneously collects high-quality data for off-policy optimization and ensures safety. Therefore, our problem is

$$\begin{aligned} &\min\ g(\pi_e) \\ &\text{s.t. } \pi_e \in \Delta_{K-1}, \min_{\bar{r}} V(\pi_e) - \alpha V(\pi_0) \geq 0\,. \end{aligned} \tag{1}$$

To instantiate the second constraint of (1), we distinguish two cases based on prior information about $\bar{r}$:

- **No Side Information:** When a brand new experiment is carried out, we have no information about $\bar{r}$. In this case, we assume no extra information about $\bar{r}$ except for being bounded, *i.e.*, $\bar{r} \in [0, 1]^K$.
- **Side Information:** Thanks to historical data from past experiments, prior information about $\bar{r}$ is often available in the form of probabilistic prior (Bastani et al., 2021b; Kveton et al., 2021; Simchowitz et al., 2021) or confidence intervals (Zhang et al., 2020). In this case, we assume that side information about $\bar{r}$ is given as confidence intervals (as this can also be constructed with a given prior), *i.e.*, $\forall a \in \mathcal{A}$, $\bar{r}(a) \in [L(a), U(a)] \subset [0, 1]$.

2.1 Mixing with Uniform is Sub-Optimal

We first show that even a simple variant of our problem has an interesting structure. Specifically, we take the *no side information* case as an example, and show that mixing of uniform exploration with the production policy is sub-optimal.

Mixing with Uniform Exploration: As indicated by its name, this heuristic would follow π_0 for $(1-\beta)$ fraction of the time while uniformly sample all the actions otherwise. Formally, the policy can be defined as $\pi_\beta := (1-\beta)\pi_0 + \beta \mathbf{1}_K/K$. This is a commonly used strategy for multi-armed bandit (see *e.g.*, Section 1.2.1 of Slivkins (2019)), reinforcement learning (see *e.g.*, Section 2.2 of Sutton and Barto (2018)), and conservative online exploration (Wu et al., 2016; Yang et al., 2021).

Balance the Amount of Exploration: Suppose w.l.o.g. that $\pi_0(1) \leq \ldots \leq \pi_0(K)$. It is easy to verify that $\pi_\beta(1) \leq \ldots \leq \pi_\beta(K)$ and $g(\pi_\beta) = \pi_\beta(1)^{-1}$. Since $\pi_0(1) \leq 1/K$, it is evident that a larger β would lead to a smaller $g(\pi_\beta)$. However, we may not be able to set $\beta = 1$ due to the safety constraint. To satisfy the safety constraint, we need to enforce that

$$\pi_\beta(a) \geq \alpha\pi_0(a) \qquad \forall a \in \mathcal{A}. \tag{2}$$

This is because if there exists an action $a \in \mathcal{A}$ such that $\pi_\beta(a) < \alpha\pi_0(a)$, then the safety constraint can be violated by setting $\bar{r}(a') = 0$ for all $a' \in \mathcal{A} \setminus \{a\}$. By solving the inequalities $(1-\beta)\pi_0(a) + \beta/K \geq \alpha\pi_0(a)$ for all a, we get

$$\beta \leq \beta_* := \min\left\{\frac{1-\alpha}{1-(K\pi_0(K))^{-1}}, 1\right\}.$$

Intuitively, this is because as we decrease β, the safety constraint is violated first for the most frequently played action. At that point, we know that $\alpha\pi_0(K) = (1-\beta_*)\pi_0(K) + \beta_*/K$.

Nevertheless, the following example shows that π_{β_*} is not always optimal.

Example 1. *Let $K = 3$, the production policy be $\pi_0 = (0.1, 0.3, 0.6)^\top$, and the safety parameter be $\alpha = 0.8$. Then $\beta_* = 0.45$ and $\pi_{\beta_*} = (0.205, 0.315, 0.48)^\top$.*

Now consider the policy $\pi = (0.26, 0.26, 0.48)^\top$. *We can verify that the safety constraint is satisfied as* $\pi(a) \geq \alpha\pi_0(a)$ *for all* $a \in \mathcal{A}$. *But we have* $g(\pi_{\beta_*}) = 0.205^{-1} > 0.26^{-1} = g(\pi)$, *and thus* π_{β_*} *is sub-optimal.*

The above example shows that mixing of the production policy with a uniform distribution yields a sub-optimal logging policy. In Appendix A of the full version (Zhu and Kveton, 2021), we show that π_β would be sub-optimal if $K > 3$ and α is above a certain threshold (*i.e.*, when the safety constraint is not too loose) while it would be optimal otherwise.

3 TABULAR SAFE OPTIMAL DESIGN

Motivated by our discussions in Section 2.1, we introduce safe optimal designs with and without side information. We start with the so-called tabular case.

3.1 Safe Optimal Design Without Side Information

We note that π_β in Section 2.1 is sub-optimal because the peeled-off probability mass from π_0 is added uniformly to all actions instead of those with the lowest probabilities, so as to reduce g maximally.

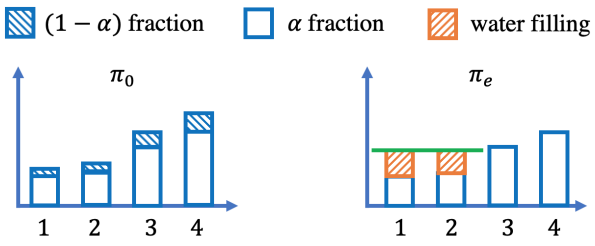

Figure 1: Water-filling method.

This motivates our *water-filling* method, which first takes $(1-\alpha)$ portion of the mass from all $\pi_0(a)$ to form $\pi'(a)$ without violating the safety constraint, *i.e.*, $\pi' = \alpha\pi_0$, and then re-allocates the peeled-off mass to π' in a greedy manner. That is, as shown in Figure 1, it successively increases the probability mass of the actions with the lowest probabilities in π_0 until all the $(1-\alpha)$ probability mass is exhausted.

Water-Filling Method: Formally, assuming w.l.o.g. that $\pi'(1) \leq \pi'(2) \leq \ldots \leq \pi'(K)$, the algorithm searches for the largest $k \in [K]$ such that $k \cdot \pi'(k) + \sum_{i=k+1}^{K} \pi'(i) \leq 1$, and then sets $\pi_e(i) = (1 - \sum_{i=k+1}^{K} \pi'(i))/k$ for all $i \leq k$ and $\pi_e(i) = \pi'(i)$ for all $i \geq k+1$. Now we establish that the water-filling method is optimal.

Theorem 1. *For any policy* π *that satisfies the safety constraint, we have* $\min_a \pi_e(a) \geq \min_a \pi(a)$ *in the no side information case.*

The complete proof of this theorem is provided in Appendix B of Zhu and Kveton (2021).

3.2 Safe Optimal Design With Side Information

Now we turn to the case with side information. The side information gives us more flexibility in satisfying the safety constraint. Notably, now $\pi_e(a) < \alpha\pi_0(a)$ can happen for some actions a as long as π_e allocates enough probability to actions with high expected rewards to compensate for this deficit. The water-filling method in Section 3.1 does not solve this problem optimally anymore. Instead, we formulate the problem of finding the optimal policy π_e as

$$\begin{aligned} \mathrm{P}_1(L, U, \pi_0): \quad & \max \gamma \\ & \text{s.t. } \pi_e \geq \gamma \mathbf{1}_K\,,\ \pi_e \in \Delta_{K-1}\,, \\ & \min_{\bar{r} \in [L,U]} (\pi_e - \alpha\pi_0)^\top \bar{r} \geq 0\,. \end{aligned}$$

Here, L, U, and π_0 are given input parameters of P_1. γ is a tight lower bound for $\min_a \pi_e$ and by maximizing γ, we equivalently minimize $g(\pi_e)$. The last constraint enforces that $V(\pi_e) \geq \alpha V(\pi_0)$ holds for all possible $\bar{r} \in [L, U] \subseteq [0,1]^K$. Note that when $[L, U] = [0,1]^K$, we can recover the solution of the water-filling method for the no side information case.

One challenge posed by $\mathrm{P}_1(L, U, \pi_0)$ is that its last constraint implicitly contains infinitely many constraints. These constraints can be satisfied incrementally using the cutting-plane method (see, *e.g.*, Chapter 6.3 of Bertsimas and Tsitsiklis (1997)). More elegantly though, motivated by robust optimization (Ben-Tal et al., 2009), we consider the following sub-optimization problem based on the last constraint

$$\begin{aligned} \mathrm{P}_2(L, U, \pi_0, \pi_e): \quad & \min\ (\pi_e - \alpha\pi_0)^\top \bar{r} \\ & \text{s.t. } L \leq \bar{r} \leq U \end{aligned}$$

and its dual

$$\begin{aligned} \mathrm{D}_2(L, U, \pi_0, \pi_e): & \max\ L^\top z_1 - U^\top z_2 \\ & \text{s.t. } z_1 - z_2 = \pi_e - \alpha\pi_0,\ z_1, z_2 \geq 0\,, \end{aligned}$$

where z_1 and z_2 are K-dimensional vectors serving as dual variables. Since $\mathrm{P}_2(L, U, \pi_0, \pi_e)$ has a finite optimal value, by strong duality (see *e.g.*, Section 4 of Bertsimas and Tsitsiklis (1997)), we have that the optimal objective values of $\mathrm{P}_2(L, U, \pi_0, \pi_e)$ and $\mathrm{D}_2(L, U, \pi_0, \pi_e)$ are the same. Thus $\mathrm{P}_1(L, U, \pi_0)$ can be equivalently written as

$$\mathrm{P}_3(L, U, \pi_0): \quad \max \gamma$$

$$\text{s.t. } \pi_e \geq \gamma \mathbf{1}_K\,,\ \pi_e \in \Delta_{K-1}\,,$$
$$L^\top z_1 - U^\top z_2 \geq 0\,,$$
$$z_1 - z_2 = \pi_e - \alpha\pi_0\,,\ z_1, z_2 \geq 0\,.$$

Intuitively, using the duality between $\mathrm{P}_2(L, U, \pi_0, \pi_e)$ and $\mathrm{D}_2(L, U, \pi_0, \pi_e)$, we translate the minimization problem in the last constraint of $\mathrm{P}_1(L, U, \pi_0)$ to a maximization problem. As a consequence, instead of checking whether π_e satisfies $(\pi_e - \alpha\pi_0)^\top \bar{r} \geq 0$ for all possible $\bar{r} \in [L, U]$, one only needs to find a single pair z_1, z_2 that satisfies the last three constraints in $\mathrm{P}_3(L, U, \pi_0)$. Therefore, $\mathrm{P}_3(L, U, \pi_0)$ is a linear program that can be solved directly.

Following the duality argument above, the equivalence of $\mathrm{P}_1(L, U, \pi_0)$ and $\mathrm{P}_3(L, U, \pi_0)$ can be established. For completeness, we include the proof of the following theorem in Appendix C of Zhu and Kveton (2021).

Theorem 2. *The optimal value of $P_1(L, U, \pi_0)$ is equal to the optimal value of $P_3(L, U, \pi_0)$.*

Remark 1. *An alternative way of solving $P_1(L, U, \pi_0)$ follows from the observation that, in the last constraint of $P_1(L, U, \pi_0)$, the minimum is attained at either $\bar{r}(a) = L(a)$ (if $\pi_e(a) - \alpha\pi_0(a) \geq 0$) or $U(a)$ (if $\pi_e(a) - \alpha\pi_0(a) < 0$). We can thus introduce a variable $z \in \mathbb{R}^K$ to serve as a coordinate-wise lower bound for $(\pi_e - \alpha\pi_0)^\top L$ and $(\pi_e - \alpha\pi_0)^\top U$, and mandate that $z^\top \mathbf{1}_K \geq 0$ to ensure $\min_{\bar{r} \in [L,U]} (\pi_e - \alpha\pi_0)^\top \bar{r} \geq 0$. Consequently, we can rewrite $P_1(L, U, \pi_0)$ as $P_1'(L, U, \pi_0)$:*

$$\max \gamma$$
$$\text{s.t. } \pi_e \geq \gamma \mathbf{1}_K\,,\ \pi_e \in \Delta_{K-1}\,,\ z^\top \mathbf{1}_K \geq 0\,,$$
$$(\pi_e - \alpha\pi_0)^\top L \geq z\,,\ (\pi_e - \alpha\pi_0)^\top U \geq z\,.$$

The equivalence of $P_1(L, U, \pi_0)$ and $P_1'(L, U, \pi_0)$ is established in Appendix D of Zhu and Kveton (2021).

3.3 Off-Policy Evaluation and Optimization

Now we apply our results to off-policy evaluation where we use data collected by our logging policy to estimate the expected reward of some policy π without deploying it. Previously, to ease presentation, we omitted the dependence on context in the reward functions. In this section, we consider a more practical contextual setting (Li et al., 2011; Dudık et al., 2014).

Additional Notations and Setup: Following Section 2, we consider the tabular action set. To model the contextual information, we assume that there is a finite set of contexts $\mathcal{X}$. A policy $\pi : \mathcal{X} \to \Delta_{K-1}$ is a mapping from a context to a probability distribution over actions, *i.e.*, $\pi(a \mid x)$ is the probability of taking action $a \in \mathcal{A}$ given context $x \in \mathcal{X}$. We assume that the random reward of choosing action a under context x is a $[0, 1]$-valued random variable with mean $\bar{r}(x, a)$. We collectively denote $\bar{r}(x, \cdot) = (\bar{r}(x, 1), \ldots, \bar{r}(x, K))^\top$ and $\bar{r} = (\bar{r}(x, \cdot))_{x \in \mathcal{X}}$. We let $\mathcal{C}$ be the distribution of the context. Let $V(\pi) = \sum_{x \in \mathcal{X}} \mathcal{C}(x) V(\pi(\cdot \mid x))$ and $V(\pi(\cdot \mid x)) = \sum_{a \in \mathcal{A}} \pi(a \mid x) \bar{r}(x, a)$ be the expected and conditional expected rewards, respectively, of policy π. With some abuse of notation, we let $\pi(\cdot \mid x) = (\pi(1 \mid x), \ldots, \pi(K \mid x))^\top$ be a vectorized policy π conditioned on context x and $g(\pi) = \max_{x \in \mathcal{X}, a \in \mathcal{A}} 1/\pi(a \mid x)$. Here, $g(\pi)$ is proportional to the maximum width of a *high-probability confidence interval* over $a \in \mathcal{A}$ and $x \in \mathcal{X}$ if we use π to collect data. We use it to measure the quality of our logging policy π_e.

Our logging policy π_e, whose expected reward is at least α of that of the production policy π_0, samples actions for n times and collects a dataset $\mathcal{D} = \{(x_t, a_t, r_t)\}_{t=1}^n$ of size n. Here $r_t \in [0, 1]$ is a stochastic reward of action a_t under context x_t in round t, with mean $\bar{r}(x_t, a_t)$.

Inverse Propensity Score (IPS) Estimator: To estimate the expected reward of any policy π from $\mathcal{D}$, we use the asymptotically optimal and unbiased IPS estimator (Rosenbaum and Rubin, 1983; Wang et al., 2017) as an example. Our IPS estimator is

$$\hat{V}(\pi) = \frac{1}{n} \sum_{t=1}^{n} \frac{\pi(a_t \mid x_t)}{\pi_e(a_t \mid x_t)} r_t\,. \tag{3}$$

Since $r_t \in [0, 1]$, we know that each individual term in the IPS estimator is $g(\pi_e)^2/4$-sub-Gaussian. Therefore, by Hoeffding's inequality (Hoeffding, 1963), for any fixed policy π, $|\hat{V}(\pi) - V(\pi)| \leq g(\pi_e)\sqrt{\log(2/\delta)/(8n)}$ holds with probability at least $1 - \delta$. Intuitively, this means that we get a better estimator of $V(\pi)$ by minimizing $g(\pi_e)$. In what follows, we show how our prior results can help here.

No Side Information: Even if we have full access to the context distribution $\mathcal{C}$, we need to enforce $V(\pi_e(\cdot \mid x)) \geq \alpha V(\pi_0(\cdot \mid x))$ across all $x \in \mathcal{X}$ to ensure $V(\pi_e) \geq \alpha V(\pi_0)$, Otherwise, suppose that there exists $x \in \mathcal{X}$ such that $V(\pi_e(\cdot \mid x)) < \alpha V(\pi_0(\cdot \mid x))$. Then one could set $\bar{r}(x', a) = 0$ for all $x' \neq x$ and a to violate the safety constraint. In this case, we implement the water-filling method for each context $x \in \mathcal{X}$ separately to minimize $\max_{a \in \mathcal{A}} 1/\pi_e(a \mid x)$, which subsequently minimizes $g(\pi_e)$ without violating the safety constraint.

Side Information: In this case, we have access to side information $\bar{r} \in [L, U]$. To further incorporate the distribution of $\mathcal{X}$, we notice that $V(\pi_e(\cdot \mid x)) < \alpha V(\pi_0(\cdot \mid x))$ could possibly occur for some x as long as π_e performs better in other contexts. To this end, we formulate the optimization jointly over all $x \in \mathcal{X}$,

i.e.,

$$\max \gamma$$
$$\text{s.t. } \pi_e \geq \gamma \mathbf{1}_{K\times|\mathcal{X}|}\,,\ \pi_e(\cdot \mid x) \in \Delta_{(K-1)}\ \forall x \in \mathcal{X}\,,$$
$$\min_{\bar{r}\in[L,U]} \sum_{x\in\mathcal{X}} \mathcal{C}(x)(\pi_e(\cdot \mid x) - \alpha\pi_0(\cdot \mid x))^\top \bar{r}(x,\cdot) \geq 0\,.$$

This optimization problem can be solved using the same duality trick as in Section 3.2. We remark that if $L = \mathbf{0}_{K\times|\mathcal{X}|}$ and $U = \mathbf{1}_{K\times|\mathcal{X}|}$, this recovers the no side information case and we get the same solution as water-filling applied separately to each context.

Performance Guarantee: Recall that $g(\pi_e)$ is exactly the minimized objective in the above optimization problem. We are now ready to show that our results for safe optimal experimental design can provide universal improvement in off-policy evaluation that further benefits the downstream optimization task.

Lemma 3. *Let $\hat{V}(\pi)$ be the IPS estimate for the value of policy π in (3). Then with probability at least $1-\delta$, $\max_\pi |\hat{V}(\pi) - V(\pi)| \leq 3g(\pi_e)\sqrt{K|\mathcal{X}|\log(n/\delta)/(4n)}$. Also let $\hat{\pi} = \arg\max_\pi \hat{V}(\pi)$ and $\pi_* = \arg\max_\pi V(\pi)$. Then $V(\pi_*) - V(\hat{\pi}) \leq 3g(\pi_e)\sqrt{K|\mathcal{X}|\log(n/\delta)/n}$ holds with probability at least $1-\delta$.*

The complete proof of Lemma 3 is provided in Appendix E of Zhu and Kveton (2021).

4 LINEAR SAFE OPTIMAL DESIGN

So far we assumed a tabular action set $\mathcal{A}$, where the expected rewards of actions are unrelated. While this setting is suitable for a small number of actions, the performance (*i.e.*, the objective function g) would quickly deteriorate if $|\mathcal{A}|$ was large. The reason is that, if no correlations exist among the expected rewards, $\min_{a\in\mathcal{A}} \pi(a) \leq 1/|\mathcal{A}|$ as $\sum_{a\in\mathcal{A}} \pi(a) = 1$, and hence $g(\pi) \geq |\mathcal{A}|$ even without any safety constraints. This essentially implies that if we apply our tabular methods to a large action set, the off-policy evaluation and optimization errors in Lemma 3 would be $\Omega(|\mathcal{A}|)$! Even worse, in practice, $|\mathcal{A}|$ is expected to be large in many popular applications, such as the large pool of ads in online advertising (Li et al., 2010; Chu et al., 2011) or the combinatorial action space in online recommendations (Swaminathan et al., 2017; McInerney et al., 2020; Vlassis et al., 2021).

To address this challenge, prior works used features. In the linear function approximation (Dani et al., 2008; Abbasi-Yadkori et al., 2011; Swaminathan et al., 2017), the assumption is that the expected reward of each action is linear in the action's features and an underlying shared reward parameter. We adopt this approach and generalize our results to the linear function approximation. Let $\mathcal{A} \subset \mathbb{R}^d$ be the action set that contains a collection of d-dimensional feature vectors with $\|a\|_2 \leq 1\ \forall a \in \mathcal{A}$. For any logging policy $\pi : \mathcal{A} \to [0,1]$, we generalize $g(\pi) \in \mathbb{R}$ in Section 2 to

$$g(\pi) = \max_{a\in\mathcal{A}} a^\top G(\pi)^{-1} a\,, \tag{4}$$

where $G(\pi) = \sum_{a\in\mathcal{A}} \pi(a) a a^\top$. We note that the tabular case is a special case where $\mathcal{A}$ is the standard Euclidean basis.

Similarly to the tabular case, $g(\pi)$ is proportional to the maximum width of a high-probability confidence interval over $a \in \mathcal{A}$ (see *e.g.*, Section 21.1 of Lattimore and Szepesvari (2018)). Our goal is to design a logging policy π_e that minimizes $g(\pi_e)$, so as to minimize our estimation error. In *absence* of the safety constraint, this is the general form of the G-optimal design (Kiefer and Wolfowitz, 1960), which is a convex optimization problem that can be solved efficiently by the Frank-Wolfe algorithm (see *e.g.*, Fedorov (1972)). To describe the safety constraint, we let $\theta_* \in \mathcal{S}_{d-1} = \{\theta \in \mathbb{R}^d : \|\theta\|_2 \leq 1\}$ be an unknown parameter vector and $\bar{r}(a) = a^\top \theta_*$ ($\in [0,1]$) be the expected reward of action a. Then the safety constraint would require that $V(\pi_e) \geq \alpha V(\pi_0)$ for all $\theta_* \in \mathcal{S}_{d-1}$.

Remark 2. *We point out that if we consider no side information (Section 3.1) or the coordinate-wise side information (Section 3.2), we can apply the results from Section 3 to compute the optimal designs.*

Side Information: In linear models, confidence regions on θ_* are often given in the form of ellipsoids (see, *e.g.*, Dani et al. (2008); Abbasi-Yadkori et al. (2011); Ban and Keskin (2020) or Chapter 20 of Lattimore and Szepesvari (2018)). We consider this generalization here, by assuming that the unknown parameter θ_* falls in a confidence ellipsoid (possibly with high probability) $\Theta := \left\{\theta \in \mathbb{R}^d : (\theta - \bar{\theta})^\top \bar{\Sigma}^{-1} (\theta - \bar{\theta}) \leq 1\right\}$. Here $\bar{\theta} \in \mathbb{R}^d$ is the center of the ellipsoid and $\bar{\Sigma}^{-1} \in \mathbb{R}^{d\times d}$ is a positive definite matrix whose eigenvectors are the principal axes of the ellipsoid and whose eigenvalues are the reciprocals of the squares of the semi-axes.

To ease exposition, we make an assumption that $\Theta \subset \mathcal{S}_{d-1}$ and define $d \times K$ matrix $A = (a^{(1)}, \ldots, a^{(K)})$, where $a^{(i)}$ is the i-th action in $\mathcal{A}$. Then the safety constraint can be written as $\pi_e^\top A^\top \theta_* \geq \alpha \pi_0^\top A^\top \theta_*\ \ \forall \theta_* \in \Theta$, and the problem of finding the optimal logging policy that satisfies the safety constraint is

$$\begin{aligned} \mathrm{P}_4(\Theta, \pi_0): \quad & \min\ g(\pi_e) \\ & \text{s.t. } \pi_e \in \Delta_{K-1}\,, \\ & \max_{\theta_*\in\Theta} (\alpha\pi_0 - \pi_e)^\top A^\top \theta_* \leq 0\,. \end{aligned}$$

As in the tabular case, the last constraint of $\mathrm{P}_4(\Theta, \pi_0)$ also requires the inequality to hold for a continuum of

θ_*, and hence implicitly consists of infinitely many constraints. We could follow the duality approach in Section 3.2, to convert $\mathrm{P}_4(\Theta, \pi_0)$ to a convex optimization problem with a quadratic constraint. However, solving it directly via conventional iterative convex optimization algorithm (*e.g.*, gradient descent) would still be computationally challenging. This is because we would need a computationally expensive projection step upon each iteration of update.

4.1 Frank-Wolfe with a Cutting Plane Method

We solve problem $\mathrm{P}_4(\Theta, \pi_0)$ without projections by using a Frank-Wolfe algorithm (Frank and Wolfe, 1956) with the cutting-plane method (see, *e.g.*, Chapter 6.3 of Bertsimas and Tsitsiklis (1997)).

The algorithm is iterative and we denote its output after iteration i by $\pi^{(i)}$ ($\pi^{(0)}$ is initialized to π_0). In each iteration, the Frank-Wolfe algorithm proceeds by minimizing a linear approximation of the objective function and sets $\pi^{(i+1)}$ to its minimizer.

More formally, we denote $H(\pi)$ as the gradient of $a^\top G(\pi)^{-1} a$ at π. Then, at the beginning of each iteration i, the Frank-Wolfe algorithm considers the following linear program

$$\begin{aligned} \mathrm{P}_6(\Theta, \pi_0): \quad \min \quad & \mathring{\pi}^\top H\left(\pi^{(i-1)}\right) \\ \text{s.t. } & \mathring{\pi} \in \Delta_{K-1}\,, \\ & \max_{\theta_* \in \Theta} (\alpha\pi_0 - \mathring{\pi})^\top A^\top \theta_* \leq 0\,. \end{aligned}$$

Let $\mathring{\pi}^{(i)}$ be the optimal solution to the above linear program. Then we set $\pi^{(i)} = \pi^{(i-1)} + \eta(\mathring{\pi}^{(i)} - \pi^{(i-1)})$, where $\eta \in [0, 1]$ is chosen such that $\max_{a \in \mathcal{A}} a^\top G(\pi^{(i)})^{-1} a$ is minimized. The final output of this algorithm is π_e.

Computing the Gradient $H(\pi)$: To work out $H(\pi)$, we compute the partial derivative of objective function w.r.t. π as

$$\begin{aligned} & \frac{\partial \max_{a \in \mathcal{A}} a^\top G\left(\pi\right)^{-1} a}{\partial \pi(a)} \\ =& a(\pi)^\top \frac{\partial G(\pi)^{-1}}{\partial \pi(a)} a(\pi) \\ =& -a(\pi)^\top G(\pi)^{-1} a a^\top G(\pi)^{-1} a(\pi) \\ =& -\left(a(\pi)^\top G(\pi)^{-1} a\right)^2\,, \end{aligned}$$

where $a(\pi) = \arg\max_{a \in \mathcal{A}} a^\top G(\pi)^{-1} a$ is the action that achieves the maximum for a given policy π. The first equality follows from the fact that $a(\pi)$ is the maximizer under π. The second equality combines the derivative of matrix inverse with the fact that $G(\pi)$ is linear in π. Consequently,

$$\begin{aligned} & H(\pi) \\ =& -\left(\left(a(\pi)^\top G(\pi)^{-1} a_1\right)^2, \ldots, \left(a(\pi)^\top G(\pi)^{-1} a_K\right)^2\right)^\top. \end{aligned}$$

Dealing with Infinitely Many Constraints: As before, the last constraint of $\mathrm{P}_6(\Theta, \pi_0)$ implicitly includes infinitely many constraints. To address this, we generate the constraints incrementally using the cutting-plane method in each iteration i. Specifically, we start with S as the empty set and denote by $\mathring{\pi}_S^{(i)}$ the corresponding optimal solution to $\mathrm{P}_6(S, \pi_0)$. For a given S, we find the most violated constraint in Θ, parameterized by

$$\begin{aligned} \theta_S =& \arg\max_{\theta_* \in \Theta} (\alpha\pi_0 - \mathring{\pi}_S^{(i)})^\top A^\top \theta_* \\ =& \bar{\theta} + \frac{\bar{\Sigma} A(\alpha\pi_0 - \mathring{\pi}_S^{(i)})}{\sqrt{(\alpha\pi_0 - \mathring{\pi}_S^{(i)})^\top A^\top \bar{\Sigma} A(\alpha\pi_0 - \mathring{\pi}_S^{(i)})}}\,. \end{aligned}$$

This closed-form solution follows from the fact that this problem is equivalent to maximizing a linear function on an ellipsoid; and we prove this in Appendix G of Zhu and Kveton (2021). Then S is updated to $S \cup \{\theta_S\}$, and we repeat this until no constraint is violated, *i.e.*, $\max_{\theta_* \in \Theta} (\alpha\pi_0 - \mathring{\pi}_S^{(i)})^\top A^\top \theta_* \leq 0$.

4.2 Off-Policy Evaluation and Optimization

Similarly to Section 3.3, we apply our results to contextual off-policy evaluation and optimization.

Additional Notations: We follow most of the notations in Section 3.3, except now the reward parameter conditioned on context x is $\theta_*(x)$ and $g(\pi) = \max_{x \in \mathcal{X}, a \in \mathcal{A}} a^\top G(\pi(\cdot \mid x))^{-1} a$. Again, $g(\pi)$ is proportional to the maximum width of a *high-probability confidence interval* over $a \in \mathcal{A}$ and $x \in \mathcal{X}$ if we use π to collect data. We use it to measure the quality of our logging policy π_e. The side information is defined as follows: for every $x \in \mathcal{X}$, $\theta_{*,x} \in \Theta_x = \{\theta \in \mathbb{R}^d : (\theta - \bar{\theta}_x)^\top \bar{\Sigma}_x^{-1} (\theta - \bar{\theta}_x) \leq 1\}$. We collectively denote $\theta_* = (\theta_{*,x})_{x \in \mathcal{X}}$ and $\Theta = (\Theta_x)_{x \in \mathcal{X}}$.

Pseudo-Inverse (PI) Estimator: To leverage the linear structure in the reward function, Swaminathan et al. (2017) proposed the PI estimator, which generalizes the IPS estimator, to estimate the expected reward of a policy π. Specifically, let $G(\pi(\cdot \mid x)) = \sum_{a \in \mathcal{A}} \pi(a \mid x) a a^\top$, the PI estimator is

$$\hat{V}(\pi) = \frac{1}{n} \sum_{t=1}^{n} r_t \left(A\pi(\cdot \mid x_t)\right)^\top G(\pi_e(\cdot \mid x_t))^{-1} a_t\,. \quad (5)$$

where $A\pi(\cdot \mid x)$ is the average action feature vector under $\pi(\cdot \mid x)$. Here we slightly overload our notation

and use G^{-1} as the pseudo-inverse of G. Swaminathan et al. (2017) showed in Proposition 1 that $\hat{V}(\pi)$ is an unbiased estimator of $V(\pi)$. From the triangle and Cauchy-Schwarz inequalities, we have that

$$\begin{aligned}
&\left|(A\pi(\cdot \mid x_t))^\top G(\pi_e(\cdot \mid x_t))^{-1} a_t\right| \\
&= \left|\sum_{a\in\mathcal{A}} \pi(a \mid x_t) a^\top G(\pi_e(\cdot \mid x_t))^{-1} a_t\right| \\
&\leq \sum_{a\in\mathcal{A}} \pi(a \mid x_t) \left|a^\top G(\pi_e(\cdot \mid x_t))^{-1} a_t\right| \\
&\leq \sum_{a\in\mathcal{A}} \pi(a \mid x_t) \sqrt{a^\top G(\pi_e(\cdot \mid x_t))^{-1} a} \\
&\quad \times \sqrt{a_t^\top G(\pi_e(\cdot \mid x_t))^{-1} a_t} \leq g(\pi_e)\,.
\end{aligned}$$

Therefore, each of the terms in the summand of (5) is $g(\pi_e)^2$-sub-Gaussian.

Side Information: To incorporate the distribution of $\mathcal{X}$ and the side information, we consider the following optimization problem

$$\begin{aligned}
&\min\ g(\pi_e) \\
&\text{s.t. } \pi_e(\cdot \mid x) \in \Delta_{(K-1)}\ \forall x \in \mathcal{X}\,, \\
&\quad \max_{\theta_*\in\Theta} \sum_{x\in\mathcal{X}} \mathcal{C}(x)(\alpha\pi_0 - \pi_e)^\top A^\top \theta_{*,x} \leq 0\,.
\end{aligned}$$

This optimization problem can be solved analogously to that in Section 4.1.

Performance Guarantee: As in the tabular case, $g(\pi_e)$ is exactly the minimized objective in the above optimization problem. We are now ready to link it to off-policy evaluation and optimization guarantees.

Lemma 4. *Let $\lambda_*(x)$ be the minimum non-zero eigenvalue of $G(\pi_e(\cdot \mid x))$, $\lambda_* = \min_{x\in\mathcal{X}} \lambda_*(x)$, and $\hat{V}(\pi)$ be the PI estimate for the value of policy π in (5). Then with probability at least $1-\delta$, $\max_\pi \left|\hat{V}(\pi) - V(\pi)\right| \leq 3g(\pi_e)\sqrt{d|\mathcal{X}|\log(n/(\delta\min\{1,\sqrt{\lambda_*}\}))/(4n)}$. Further, let $\hat{\pi} = \arg\max_\pi \hat{V}(\pi)$ and $\pi_* = \arg\max_\pi V(\pi)$. Then with probability at least $1-\delta$, we have that $V(\pi_*) - V(\hat{\pi}) \leq 3g(\pi_e)\sqrt{d|\mathcal{X}|\log(n/(\delta\min\{1,\sqrt{\lambda_*}\}))/n}$.*

The complete proof of Lemma 4 is provided in Appendix F of Zhu and Kveton (2021).

5 EXPERIMENTS

We conduct two experiments. In Section 5.1, we illustrate the basic properties of our approach on a simple example. We evaluate it on a diverse set of problems in Section 5.2.

Our approach is implemented as described in Section 4.1 and we call it `SafeOD`, which is an abbreviation for safe optimal design. We compare it with two baselines. The first baseline is the G-optimal design π_g. The G-optimal design can be viewed as an unsafe variant of `SafeOD`, *i.e.*, $\alpha = 0$. The second baseline is a mixture policy $\pi_{\text{mix}} = \alpha\pi_0 + (1-\alpha)\mathbf{1}_K/K$. This policy is guaranteed to satisfy the safety constraint but may not maximize information gain.

All logging policies π are evaluated by three criteria. The first is the *design width* $\sqrt{g(\pi)}$, which is defined in (4) and reflects how well π minimizes uncertainty over all actions. Lower values are better. The second criterion is the *safety violation* $\max_{\theta_*\in\Theta}(\alpha\pi_0 - \pi)^\top A^\top\theta_*$ (Section 4), which measures how much π violates the safety constraint for being close to the production policy π_0. Lower values are better. The last metric is *off-policy gap*, which measures the suboptimality of the best off-policy estimated action on data collected by π. This metric is computed as follows. First, we drawn $\theta_* \in \Theta$, uniformly at random, and find the best action a_* under θ_*. Second, we collect a dataset $\mathcal{D}$ of size $n = 10d$, were the noisy observation of action a is $a^\top\theta_* + \varepsilon$ for $\varepsilon \sim \mathcal{N}(0,1)$. Finally, we compute the MLE of θ_* from $\mathcal{D}$, which we denote by $\hat{\theta}$, and find the best action $\hat{a}$ under $\hat{\theta}$. The off-policy gap is $(a_* - \hat{a})^\top\theta_*$ and we estimate it from 1 000 random runs for any given logging policy, as described above.

5.1 Illustrative Example

We start with $\mathcal{A} = \{(1,0),(0,1)\}$, $\pi_0 = (0.2, 0.8)$, and $\alpha = 0.9$; and Θ is given by $\bar{\theta} = (1,2)$ and $\bar{\Sigma} = 0.1 I_2$. In this case, π_0 takes the most rewarding action with a high probability of 0.8. Therefore, `SafeOD` cannot differ much from π_0 and is $\pi_e = (0.330, 0.670)$. This design satisfies the safety constraint and its width is 1.74. In comparison, the G-optimal design is $\pi_g = (0.5, 0.5)$ and obviously violates the safety constraint. For instance, even at $\bar{\theta}$, the constraint violation is $0.9 \cdot (0.2 \cdot 1 + 0.8 \cdot 2) - 0.5 \cdot 3 = 0.12$. However, its width is only 1.414. The mixture policy π_{mix} satisfies the safety constraint but its width is 2.085, about 15% higher than in `SafeOD`.

Next we set $\bar{\theta} = (2,1)$. In this case, π_0 takes the least rewarding action with a high probability of 0.8. Therefore, `SafeOD` can depart significantly from π_0 and is $\pi_e = (0.5, 0.5)$. This design satisfies the safety constraint and its width is 1.141. The G-optimal design coincides with `SafeOD` ($\pi_g = \pi_e$). The mixture policy π_{mix} also satisfies the safety constraint but its width is 2.085.

In conclusion, `SafeOD` combines the best properties of π_g and π_{mix}. When the safety constraint is strict, `SafeOD` satisfies it. When it is not, `SafeOD` has a low width, similarly to the G-optimal design.

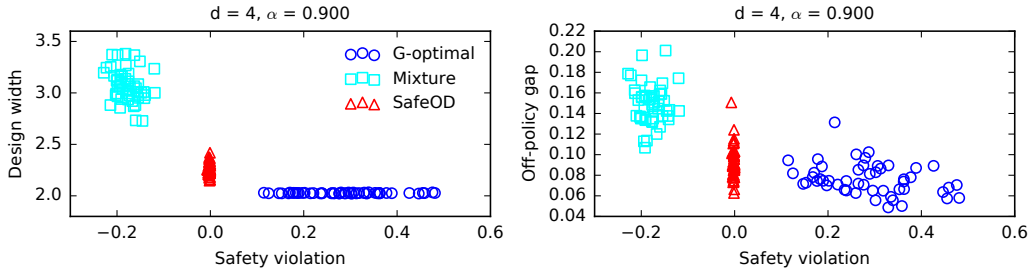

Figure 2: Comparison of `SafeOD` to the G-optimal optimal design and the mixture policy.

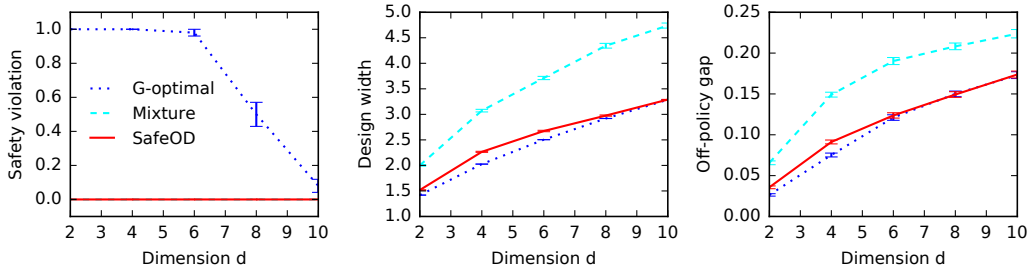

Figure 3: Comparison of `SafeOD` to the G-optimal optimal design and the mixture policy. We fix the safety parameter at $\alpha = 0.9$ and vary d. Each safety violation is the fraction of violated safety constraints in 50 runs.

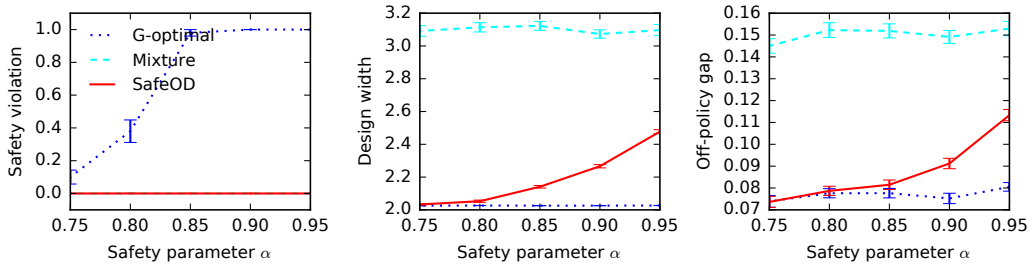

Figure 4: Comparison of `SafeOD` to the G-optimal optimal design and the mixture policy. We fix $d = 4$ and vary the safety parameter α. Each safety violation is the fraction of violated safety constraints in 50 runs.

5.2 Synthetic Problems

We also experiment with the following randomly generated problems. The number of actions is $K = 100$ and their feature vectors are drawn uniformly from a d-dimensional unit sphere. The production policy π_0 is drawn uniformly from a $(K-1)$-dimensional simplex. The set Θ is defined by $\bar{\Sigma} = I_d$ and $\bar{\theta}$, which is drawn uniformly from a d-dimensional hypercube $[1, 2]^d$. We vary d and α, and generate 50 random problems for each setting.

In Figure 2, we report results for $d = 4$ and $\alpha = 0.9$. We observe that the G-optimal designs have low widths but violate the safety constraint. The mixture policy always satisfies the safety constraint but leads to high design widths. `SafeOD` strikes the balance between the two objectives, by minimizing the design width under the safety constraint. In all cases, design widths correlate with off-policy gaps.

In Figure 3, we fix $\alpha = 0.9$ and vary d; while in Figure 4, we fix $d = 4$ and vary α. In general, we observe that `SafeOD` performs similarly to the G-optimal design whenever the safety constraint is easy to satisfy, when the number of features d is large or the safety parameter α is small. In all other cases, `SafeOD` produces designs of higher widths and off-policy gaps in return for satisfying the safety constraint. The mixture policy always satisfies the safety constraint but leads to high design widths and off-policy gaps.

6 CONCLUSIONS

In this work, we design safe optimal logging policies that simultaneously collect high-quality data for off-policy learning and achieve competitive expected rewards to a production policy. We first show that the policy induced by mixing the production policy and uniform exploration is safe but sub-optimal in general. Then we develop optimal solutions for various setting, and relate them to off-policy evaluation and optimization. Finally, we conduct extensive numerical experiments to demonstrate the performance of our proposed designs.

References

Yasin Abbasi-Yadkori, David Pál, and Csaba. Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances Neural Information Processing Systems 25 (NIPS)*, 2011.

Jacob Abernethy, Kareem Amin, and Ruihao Zhu. Threshold bandits, with and without censored feedback. In *Advances in Neural Information Processing Systems 29 (NIPS)*, 2016.

Jean-Yves Audibert, Sebastien Bubeck, and Remi Munos. Best arm identification in multi-armed bandits. *Proceedings of the 23th Conference on Learning Theory*, 2010.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. In *Machine learning, 47, 235–256*, 2002.

Mohammad Javad Azizi, Branislav Kveton, and Mohammad Ghavamzadeh. Fixed-budget best-arm identification in contextual bandits: A static-adaptive algorithm. *arXiv:2106.04763v6 [cs.LG]*, 2021.

Gah-Yi Ban and N. Bora Keskin. Personalized dynamic pricing with machine learning: High dimensional features and heterogeneous elasticity. In *Management Science (Forthcoming)*, 2020.

Hamsa Bastani, Pavithra Harsha, Georgia Perakis, and Divya Singhvi. Learning personalized product recommendations with customer disengagement. *Manufacturing & Service Operations Management*, 2021a.

Hamsa Bastani, David Simchi-Levi, and Ruihao Zhu. Meta dynamic pricing: Learning across experiments. *Management Science*, 2021b.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Optimal exploration-exploitation in a multi-armed-bandit problem with non-stationary rewards. In *Forthcomming in Stochastic Systems*, 2018.

Sebastien Bubeck, Remi Munos, and Gilles Stoltz. Pure exploration for multi-armed bandit problems. *arXiv:0802.2655v6*, 2010.

Boxiao Chen, Xiuli Chao, and Yining Wang. Data-based dynamic pricing and inventory control with censored demand and limited price changes. *Operations Research 68(5): 1445-1456*, 2020.

Shouyuan Chen, Tian Lin, Irwin King, Michael R. Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. *Advances in Neural Information Processing Systems 27 (NIPS)*, 2014.

Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Hedging the drift: Learning to optimize under non-stationarity. In *arXiv:1903.01461*, 2019. URL https://arxiv.org/abs/1903.01461.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

Varsha Dani, Thomas Hayes, and Sham Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Conference on Learning Theory (COLT)*, 2008.

Lina Danilchik. Sequential a/b testing vs multi-armed bandit testing. *SplitMetrics App Growth Blog*, 2020.

Miroslav Dudık, Dumitru Erhan, John Langford, and Lihong Li. Doubly robust policy evaluation and optimization. In *Statistical Science*, 2014.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Pac bounds for multi-armed bandit and markov decision processes. *Computational Learning Theory*, 2002.

Valerii Fedorov. *Theory of Optimal Experiments Designs*. Academic Press, 1972.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. In *Naval Research Logistics Quarterly*, volume 3, pages 95–110, March - June 1956.

Amy Gallo. A refresher on a/b testing. *Harvard Business Review*, 2017.

Google Optimize. Online, 2021. URL https://marketingplatform.google.com/about/optimize/. [Last accessed September 13, 2021].

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *Journal of the American statistical association*, volume 58, pages 13–30. Taylor & Francis Group, 1963.

Kevin Jamieson and Robert Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. *Annual Conference on Information Sciences and Systems (CISS)*, 2014.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Abbas Kazerouni, Mohammad Ghavamzadeh, Yasin Abbasi Yadkori, and Benjamin Van Roy. Conservative contextual linear bandits. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jack Kiefer and Jacob Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12(5):363–366, 1960.

Branislav Kveton, Mikhail Konobeev, Manzil Zaheer, Chih wei Hsu, Martin Mladenov, Craig Boutilier, and Csaba Szepesvari. Meta thompson sampling. *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Romain Laroche, Paul Trichelair, and Remi Tachet Des Combes. Safe policy improvement with baseline bootstrapping. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3652–3661, 2019.

Tor Lattimore and Csaba Szepesvari. *Bandit Algorithms*. Cambridge University Press, 2018.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Chang Li, Branislav Kveton, Tor Lattimore, Ilya Markov, Maarten de Rijke, Csaba Szepesvári, and Masrour Zoghi. Bubblerank: Safe online learning to re-rank via implicit click feedback. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, pages 196–206, 2019.

Lihong Li, Wei Chu, John Langford, and Robert Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of International conference on World wide web (WWW)*, 2010.

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. *Proceedings of the fourth ACM International Conference on Web Search and Data Mining (WSDM)*, 2011.

James McInerney, Brian Brost, Praveen Chandar, Rishabh Mehrotra, and Ben Carterette. Counterfactual evaluation of slate recommendations with sequential reward interactions. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

Xinkun Nie, Xiaoying Tian, Jonathan Taylor, and James Zou. Why adaptively collected data have negative bias and how to correct for it. *Harvard Business Review*, 2017.

Christopher Olah. Visualizing mnist: An exploration of dimensionality reduction. *Available at https://colah.github.io/posts/2014-10-Visualizing-MNIST/*, 2014.

Optimizely. Online, 2021a. URL https://www.optimizely.com/. [Last accessed September 13, 2021].

Optimizely. A/b test. *Optipedia*, 2021b. URL https://www.optimizely.com/optimization-glossary/ab-testing/.

Francesco Orabona. A modern introduction to online learning. *arXiv:1912.13213v4 [cs.LG]*, 2019.

Chao Qin, Diego Klabjan, and Dan Russo. Improving the expected improvement algorithm. *Conference on Neural Information Processing Systems*, 2017.

Paul Rosenbaum and Donald Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 1983.

Yufei Ruan, Jiaqi Yang, and Yuan Zhou. Linear bandits with limited adaptivity and learning distributional optimal design. *Proceedings of 53rd ACM Symposium on Theory of Computing (STOC)*, 2021.

Eric M. Schwartz, Eric T. Bradlow, and Peter S. Fader. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science 36(4): 500-522*, 2017.

Jaehyeok Shin, Aaditya Ramdas, and Alessandro Rinaldo. Are sample means in multi-armed bandits positively or negatively biased? *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

David Simchi-Levi and Yunzong Xu. Phase transitions in bandits with switching constraints. *arXiv:1905.10825v4 [cs.LG]*, 2021.

Max Simchowitz, Christopher Tosh, Akshay Krishnamurthy, Daniel Hsu, Miroslav Dudik Thodoris Lykouris, and Robert E. Schapire. Bayesian decision-making under misspecified priors with applications to meta-learning. *arXiv:2107.01509v1 [cs.LG]*, 2021.

Aleksandrs Slivkins. *Introduction to Multi-Armed Bandits*. Foundations and Trends in Machine Learning, 2019.

Marta Soare, Alessandro Lazaric, and Remi Munos. Best-arm identification in linear bandits. *Advances in Neural Information Processing Systems 27 (NIPS)*, 2014.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miroslav Dudik, John Langford, Damien Jose, and Imed Zitouni. Off-policy evaluation for

slate recommendation. *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2380–2388, 2015.

Michal Valko, Alexandra Carpentier, and Remi Munos. Stochastic simultaneous optimistic optimization. *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

Nikos Vlassis, Ashok Chandrashekar, Fernando Amat Gil, and Nathan Kallus. Control variates for slate off-policy evaluation. *arXiv:2106.07914 [cs.LG]*, 2021.

Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudik. Optimal and adaptive off-policy evaluation in contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Yifan Wu, Roshan Shariff, Tor Lattimore, and Csaba Szepesvari. Conservative bandits. *International Conference on Machine Learning (ICML)*, 2016.

Junwen Yang and Vincent Y. F. Tan. Towards minimax optimal best arm identification in linear bandits. *arXiv:2105.13017v1*, 2021.

Yunchang Yang, Tianhao Wu, Han Zhong, Evrard Garcelon, Matteo Pirotta, Alessandro Lazaric, Liwei Wang, and Simon S. Du. A unified framework for conservative exploration. *arXiv:2106.11692v1 [cs.LG]*, 2021.

Kelly W. Zhang, Lucas Janson, and Susan A. Murphy. Inference for batched bandits. In *34th Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Ruihao Zhu and Branislav Kveton. Safe optimal design with applications in policy learning. *Technical Report*, 2021. URL https://arxiv.org/abs/2111.04835.