Incorporating Behavioral Constraints in Online AI Systems

Avinash Balakrishnan,[§] Djallel Bouneffouf,[§] Nicholas Mattei,[†] Francesca Rossi[§]

[§] **IBM Research**
Yorktown Heights, NY, USA
{avinash.bala,djallel.bouneffouf,francesca.rossi2}@ibm.com

[†] **Tulane University**
New Orleans, LA, USA
nsmattei@tulane.edu

Abstract

AI systems that learn through reward feedback about the actions they take are increasingly deployed in domains that have significant impact on our daily life. However, in many cases the online rewards should not be the only guiding criteria, as there are additional constraints and/or priorities imposed by regulations, values, preferences, or ethical principles. We detail a novel online agent that learns a set of behavioral constraints by observation and uses these learned constraints as a guide when making decisions in an online setting while still being reactive to reward feedback. To define this agent, we propose to adopt a novel extension to the classical contextual multi-armed bandit setting and we provide a new algorithm called Behavior Constrained Thompson Sampling (BCTS) that allows for online learning while obeying exogenous constraints. Our agent learns a constrained policy that implements the observed behavioral constraints demonstrated by a teacher agent, and then uses this constrained policy to guide the reward-based online exploration and exploitation. We characterize the upper bound on the expected regret of the contextual bandit algorithm that underlies our agent and provide a case study with real world data in two application domains. Our experiments show that the designed agent is able to act within the set of behavior constraints without significantly degrading its overall reward performance.

1 Introduction

In online decision settings, an agent must select one out of several possible actions, e.g., recommending a movie to a particular user, or proposing a treatment to a patient in a clinical trial. Each of these actions is associated with a context, e.g., a user profile, and a feedback signal, e.g., the reward or rating, is only observed for the chosen option. In these online decision settings the agent must learn the inherent trade-off between exploration, which involves identifying and understanding the reward from an action, and exploitation, which means gathering as much reward as possible from an action. We consider cases where the behavior of the online agent may need to be restricted in its choice of an action for a given context by laws, values, preferences, or ethical principles (Russell, Dewey, and Tegmark 2015). More precisely, we apply a set of *behavioral constraints* to the agent that are independent of the reward function. For instance, a parent or guardian group may want a movie recommender system (the agen)t to not recommend certain types of movies to children, even if the recommendation of such movies could lead to a high reward (Balakrishnan et al. 2018). In clinical settings, a doctor may want its diagnosis support system to not recommend a drug that typically works because of considerations related to the patients' quality of life.

Many decision problems where an agent is responsive to online feedback are modeled as a *multi-armed bandit (MAB)* problem (Mary, Gaudel, and Preux 2015; Villar, Bowden, and Wason 2015). In the MAB setting there are K *arms*, each associated with a fixed but unknown reward probability distribution (Lai and Robbins 1985; Auer, Cesa-Bianchi, and Fischer 2002). At each time step, an agent plays an arm, i.e., recommends an item to a user, and receives a reward that follows the selected arm's probability distribution, independent of the previous actions. A popular generalization of MAB is the contextual multi-armed bandit (CMAB) problem where the agent observes a d-dimensional *feature vector*, or *context*, to use along with the rewards of the arms played in the past in order to choose an arm to play. Over time, the agent learns the relationship between contexts and rewards and select the best arm (Agrawal and Goyal 2013).

In giving the agent a set of behavioral constraints, we consider the case where only *examples* of the correct behaviors are given by a teacher agent, and our online agent must learn and respect these constraints in the later phases of decision making: As an example, a parent may give examples of movies that their children can watch (or that they cannot watch) when setting up a new movie account for them.

The idea of teaching machines right from wrong has become an important research topic in both AI (Yu et al. 2018) and in other disciplines (Wallach and Allen 2008). Much of the research at the intersection of artificial intelligence and ethics falls under the heading of *machine ethics*, i.e., adding ethics and/or constraints to a particular system's decision making process (Anderson and Anderson 2011). One very important task in machine ethics is *value alignment*, i.e., the idea that an agent can only pursue goals that are aligned to human values and are therefore beneficial to humans (Russell, Dewey, and Tegmark 2015; Loreggia et al. 2018a; Loreggia et al. 2018b). However, this still leaves open the question of how to provide the values, or the behavioral constraints derived from the values, to the agent. A popular technique is called the *bottom up approach*, i.e., teaching a machine what is right and wrong by example (Allen, Smit, and Wallach 2005). We adopt this technique in our paper, since we consider the case where only *examples* of both correct and incorrect behavior are given to the agent, that must learn from these. Note that having only examples of good and bad behavior means that we need to learn the constraints from such examples. On the contrary, if the constraints were known and we received feedback on them during the recommendation phase, then we could use the stochastic combinatorial semi-bandit setting (Kveton et al. 2014). Also, if the constraints were known and budget-like, then we could use bandits with knapsacks (Agrawal and Goyal 2016). However, the case where we are only given examples of the behavioral constraints to guide the online agent's behavior does not seem to be covered by the current literature.

To give flexibility to our agent, we let the system designer to decide how much the guidelines given by the behavioral constraints should weigh on the decision of the agent during the online phase. So, to control the tradeoff between following the learned behavioral constraints and pursuing a greedy online-only policy, we expose a parameter of the algorithm called σ_{online}. This parameter allows the system designer to smoothly transition between the two policy extremes, where $\sigma_{online} = 0.0$ means that we are only following the learned constraints and are insensitive to the online reward, while $\sigma_{online} = 1.0$ means we are only following the online rewards and not giving any weight to the learned constraints.

Contributions. We propose a novel extension of the contextual bandit setting that we call the *Behavior Constrained Contextual Bandits Problem (BCCBP)*, where the agent is constrained by a policy that it has learned from observing examples of good and bad behavior. We provide a new algorithm for this setting, that we call *Behavior Constrained Thompson Sampling (BCTS)*, that is able to trade-off between the constrained behavior learned from examples given by a teaching agent and reward-driven behavior learned during online recommendation. We prove an upper bound on the expected regret of this new algorithm and evaluate it empirically on data from two real world settings. The experimental evaluation shows that it is possible to learn and act in a constrained manner while not significantly degrading the performance. This work is part of a broader research program to incorporate ethics and other constraints into artificial agents (Rossi and Mattei 2019).

2 Background and Related Work

Multi-armed Bandits. The classic multi-armed bandit (MAB) problem is a model of the trade-off between exploration and exploitation, where an agent acting in a live, online environment wants to pick, within a finite set of decisions, the one maximizing the cumulative reward (Lai and Robbins 1985; Auer, Cesa-Bianchi, and Fischer 2002). The MAB problem is a classic example of reinforcement learning, where the online agent receives signals that are then used to update their behavior. Reinforcement learning, and the MAB problem specifically, have been used to successfully solve a number of real-world problems (Sutton and Barto 2017; Mnih et al. 2013).

The contextual multi-armed bandit (CMAB) problem, a generalization of the MAB problem that exploits the presence of features for users of the online decision system, has been studied in multiple domains including recommender systems and online advertising. Optimal solutions have been provided by using a stochastic formulation (Lai and Robbins 1985; Auer, Cesa-Bianchi, and Fischer 2002), a Bayesian formulation (Thompson 1933; Kaufmann, Korda, and Munos 2012; Agrawal and Goyal 2012), and an adversarial formulation (Auer and Cesa-Bianchi 1998; Auer et al. 2002). Both LINUCB (Li et al. 2010; Chu et al. 2011) and Contextual Thompson Sampling (CTS)(Agrawal and Goyal 2013) assume a linear dependency between the expected reward of an action and its context; the representation space is modeled using a set of linear predictors and provide bounds on the expected regret.

The literature shows some recent work on combining the contextual bandit formalism with constraints. (Wu et al. 2015) propose a contextual bandits with budget and time constraints, that are expressed over the number of times, and time period, that an arm can be pulled. Such coupling effects make it difficult to obtain oracle solutions that assume known statistics of bandits. The authors develop an approximation of the oracle, referred to as Adaptive-Linear-Programming (ALP), which achieves near-optimality and only requires the ordering of expected rewards.

Additionally, Agrawal and Goyal (2016) consider multifaceted budget constraints where each arm pull exhausts some facet of a budget that needs to be optimized under. Bouneffouf et al. (2017) consider a novel formulation of the contextual bandit problem when there are constraints on the context, i.e., where only a limited number of features can be accessed by the learner at each iteration. This novel formulation is motivated by different online problems arising in clinical trials and recommender systems where accessing all parts of the context could be costly. None of these formalisms capture the setting we consider in this paper, as the budgets or time constraints are explicitly supplied a priori.

There are also a number of closely related bandit formalisms including matroid bandits, also called stochastic combinatorial semi-bandits, contextual bandits with history, and conservative bandits. (Kveton et al. 2014) consider matroid bandits which are able to optimize combinatorial functions that are expressible as matroids, e.g., linear combinations of objectives. In relation to our work, if the constraints were known to the agent, then we could leverage these algorithms. However, we assume that our agent does not receive the constraints explicitly during the constraint learning phase and does not receive feedback on the constraints during the recommendation phase. We believe that in many settings we may not have access to the constraints as an explicit set of rules or a function, rather we may only have access to (positive and negative) examples of constrained behavior from a doctor, or a set of (good and bad) decisions made by a parent, and that this ”teaching” figure may not always be able to provide feedback on each recommendation.

Shivaswamy and Joachims (2012) introduces the setting called contextual bandit with history: an agent is furnished with examples of past behavior over the same reward function and leverages these observations to guide its future behavior. In our setting we instead have two separate reward functions, the constraints and the online reward, and hence the results are not directly applicable.

Finally, Wu et al. (2016) introduces conservative bandits, which attempt to maximize the achieved reward of an online agent while keeping the cumulative reward above a certain threshold. These setting is useful, for example, when one wants to try out new advertisements but does not want revenue to fall below a certain threshold. This work is fundamentally different from our own as we want the agent to not exploit certain arms at all rather than maintain a minimum of overall reward.

Constrained and Ethical Decision Making. Humans often constrain the decisions that they take according to a number of exogenous priorities, derived by moral, ethical, religious, or business values (Sen 1974). Constrained or ethical decision making has been studied in a variety of contexts in computer science with most of the work focused on teaching a computer system to act within guidelines (Bonnefon, Shariff, and Rahwan 2016). Broadly, our work fits into constrained reinforcement learning / Safe RL (Leike et al. 2017) and value alignment (Russell, Dewey, and Tegmark 2015; Loreggia et al. 2018a; Loreggia et al. 2018b).

For example, Briggs and Scheutz (2015) discusses a rule-based system applied to scenarios in which a robot should infer that a directive leads to undesirable behavior. In this system, given some instructions, the agent first reasons about a set of conditions including “Do I know how to accomplish the task?”, and “Does accomplishing this task violate any normative principles?”. Each of these conditions is formalized as a logical expression, along with inference rules that enable the agent to infer which directives to reject.

Having the agent learn about its ethical objective function while making decisions based on this objective function is a challenging problem. In (Armstrong 2015) the authors consider this problem by exploring the consequences of an agent that uses Bayesian learning to update its beliefs about the “true” ethical objective function. At each time step, the agent makes decisions that maximize a utility function based on the agents beliefs about the ethical objective function. In reinforcement learning, Markov decision processes have been used to study both ethics and ethical decision making. In (Abel, MacGlashan, and Littman 2016) the authors argue that the reinforcement learning framework achieves the appropriate generality required to theorize about an idealized ethical artificial agent.

None of these previously proposed approaches to ethical or behavior-constrained decision making leverage the contextual bandit setting. We instead feel that the bandit setting, which has broad application across computer science and decision making, is an ideal formalism for online decision making and can be fruitfully extended to include behavioral constraints.

3 Preliminaries

Following Langford and Zhang (2008), the contextual bandit problem is defined as follows. At each time $t \in \{1, ..., T\}$, a player is presented with a *context vector* $c(t) \in \mathbf{R}^d$ and must choose an arm $k \in K = \{1, ..., |K|\}$. Let $\mathbf{r} = (r_1(t), ..., r_K(t))$ denote a reward vector, where $r_k(t) \in [0, 1]$ is a reward at time t associated with the arm $k \in A$. We assume that the expected reward is a linear function of the context, i.e. $E[r_k(t)|c(t)] = \mu_k^T c(t)$, where μ_k is an unknown weight vector (to be learned from the data) associated with arm k.

Algorithm 1 Contextual Thompson Sampling Algorithm

```
Initialize: $B = I_d, \hat{\mu} = 0_d, f = 0_d$.
Foreach $t = 1, 2, ..., T$ do
    Sample $\tilde{\mu_k}(t)$ from the $N(\hat{\mu}_k, v^2 B_k^{-1})$ distribution.
    Play arm $k_t = \underset{k \in K}{argmax}\ c(t)^\top \tilde{\mu_k}(t)$
    Observe $r_k(t)$
$B_k = B_k + c(t)c(t)^T$, $f = f + c(t)r_k(t)$, $\hat{\mu_k} = B_k^{-1} f$
End
```

The purpose of a contextual bandit algorithm A is to minimize the cumulative regret. Let $H : C \rightarrow [K]$ where C is the set of possible contexts and $c(t)$ is the context at time t, $h_t \in H$ a hypothesis computed by the algorithm A at time t and $h_t^* = \underset{h_t \in H}{\operatorname{argmax}}\ r_{h_t(c(t))}(t)$ the optimal hypothesis at the same round. The cumulative regret is: $R(T) = \sum_{t=1}^{T} r_{h_t^*(c(t))}(t) - r_{h_t(c(t))}(t)$.

One widely used way to solve the contextual bandit problem is the Contextual Thompson Sampling algorithm (CTS) (Agrawal and Goyal 2013) given as Algorithm 1. In CTS, the reward $r_k(t)$ for choosing arm k at time t follows a parametric likelihood function $Pr(r(t)|\tilde{\mu})$. Following (Agrawal and Goyal 2013), the posterior distribution at time $t + 1$, $Pr(\tilde{\mu}|r(t)) \propto Pr(r(t)|\tilde{\mu})Pr(\tilde{\mu})$ is given by a multivariate Gaussian distribution $\mathcal{N}(\hat{\mu_k}(t + 1),\ v^2 B_k(t + 1)^{-1})$, where $B_k(t) = I_d + \sum_{\tau=1}^{t-1} c(\tau)c(\tau)^\top$, d is the size of the context vectors c, $v = R\sqrt{\frac{24}{z} d \cdot ln(\frac{1}{\gamma})}$ and we have $R > 0$, $z \in [0, 1]$, $\gamma \in [0, 1]$ constants, and $\hat{\mu}(t) = B_k(t)^{-1}(\sum_{\tau=1}^{t-1} c(\tau)r_k(\tau))$.

Every step t consists of generating a d-dimensional sample $\tilde{\mu_k}(t)$ from $\mathcal{N}(\hat{\mu_k}(t),\ v^2 B_k(t)^{-1})$ for each arm. We then decide which arm k to pull by solving for $argmax_{k \in K} c(t)^\top \tilde{\mu_k}(t)$. This means that at each time step we are selecting the arm that we expect to maximize the observed reward given a sample of our current beliefs over the distribution of rewards, $c(t)^\top \tilde{\mu_k}(t)$. We then observe the actual reward of pulling arm k, $r_k(t)$ and update our beliefs.

Definition 1 (Optimal Policy)**.** *The optimal policy for solving the contextual MAB is selecting the arms at time t : $k(t) = \underset{k \in K}{argmax}\ \tilde{\mu}_k^*(t)^\top c(t)$, where $\tilde{\mu}_k^*$ the optimal mean vector for the reward driven policy.*

4 Behavior Constrained Contextual Bandits

Here we define a new type of a bandit problem, the *Behavior Constrained Contextual Bandits (BCCB)*, present a novel algorithm and agent for solving this problem, and derive an upper bound for both the expected online regret and the regret as a function of the number of constraint examples given by the teacher.

In this setting, the agent first goes through a *constraint learning phase* where it is allowed to query the user N times and receive feedback $r^e_k(t) \in [0,1]$ about whether or not the chosen decision is allowed under the desired constrained behavior during the recommendation phase. During the *online recommendation phase*, the goal of the agent is to maximize both $r_k(t) \in [0,1]$, the reward of the action k at time t, while minimizing the (unobserved) $r^e_k(t) \in [0,1]$, which models whether or not the pulling of arm k violates the behavioral constraints. During the recommendation phase, the agent receives no feedback on the value of $r^e_k(t)$, as the labeler may not be around to always provide this feedback. In order to follow and subsequently maximize alignment with the behavioral constraints the agent must learn an effective policy that implements the constraints during the first phase and apply it during the recommendation phase. In the second phase we are interested in the total *behavioral error* incurred by the agent, i.e., $E(T) = \sum_{t=1}^{T} r^e_{h_t(c(t))}(t)$.

Behavior Constrained Thompson Sampling

Our agent employs an extension of the classical Thompson sampling algorithm to first learn a constrained policy from observation and then use this constrained policy to guide the online learning task. Behavior Constrained Thompson Sampling (BCTS) contains two parts:

- **Constraint Learning Phase**: During this phase the agent learns the behavioral constraints through interaction with a teaching agent that is able to, at each iteration, provide a context to our agent as well as feedback as to whether or not the action chosen by our agent is allowed. This feedback takes the form of a binary reward revealed for the arm k that is chosen at time t, $r^e_k(t) \in [0,1]$. We use the classical Thompson sampling (CTS) algorithm (see Algorithm 2 line 2) in order to explore the constraint space and learn a policy. We also compare this against the setting where the agent chooses random arms during the teaching phase. We call the behavior that the agent learns during this phase the constrained policy μ^e where $\hat{\mu}^e_k$ and $\hat{B}_k$ denotes respectively the learned mean vector and the covariance matrix for each arm k.
- **Online Recommendation Phase**: During the recommendation phase the agent continually faces a dilemma: follow the constrained policy or follow the reward signal. Our algorithm uses the Thompson sampling strategy to estimate the expected rewards of the online policy for each arm (Algorithm 2 line 4–7), while using the constrained policy to estimate the expected behavior $\tilde{\mu_k}(t)$. It then computes a weighted combination of $\tilde{\mu}^e_k(t)$ and $\tilde{\mu_k}(t)$ for each arm using σ_{online} as weight given by the user (line 15), this weight balances between following a reward driven policy and constrained policy. It then obtains the reward (line 16) and updates the parameters of the distribution for each $\hat{\mu_k}$ (line 17). Finally, the reward $r_k(t)$ for the chosen arm k is observed, and relevant parameters are updated.

Algorithm 2 Behavior Constrained Thompson Sampling

```
 1: Initialize: $\forall k \in K, B_k = I_d$, $V_{\tilde{\mu}} = 0_d$, $V_{\mu^e} = 0_d$,
    $\hat{\mu_k} = 0_d, g_k = 0_d, \sigma_{online}$.
 2: // Constraint Learning Phase
 3: Foreach $t = 1, 2, ..., N$ do
 4:     $c(t)$ is revealed to the agent.
 5:     The agent chooses an action $k$.
 6:     The teaching agent reveals reward $r^e_k(t)$.
 7:     The agent updates its policy $\mu^e_t$.
 8:     $t = t + 1$
 9: // Online Recommendation Phase
10: Foreach $t = 1, 2, ..., T$ do
11:     Observe the context vector $c(t)$ of features.
12:     Foreach arm $k \in K$ do
13:         Sample $\tilde{\mu_k}(t)$ from $N(\hat{\mu_k}, v^2 B_k^{-1})$ distribution.
14:         Sample $\tilde{\mu}^e_k(t)$ from $N(\hat{\mu}^e_k, v^2 \hat{B}_k^{-1})$ distribution.
15:     End do
16:     Select $k(t) = \underset{k \in K}{argmax}\ \sigma_{online} \cdot \tilde{\mu_k}(t)^\top c(t) + (1 -$
    $\sigma_{online})\tilde{\mu}^e_k(t)^\top c(t)$
17:     Observe $r_k(t)$
18:     $B_k = B_k + c(t)c(t)^T$
19:     $g_k = g_k + c(t)r_k(t)$
20:     $\hat{\mu_k} = B_k^{-1} g_k$
21: End do
```

We derive an upper bound on the regret $R(T)$ of the policy computed by BCTS. We use the standard definition of the optimal policy from the Contextual MAB literature. Note that this definition is used for most bandit problems. We are also interested in studying the the behavioral error $E(T)$ incurred by the agent for any policy it may follow in the online phase.

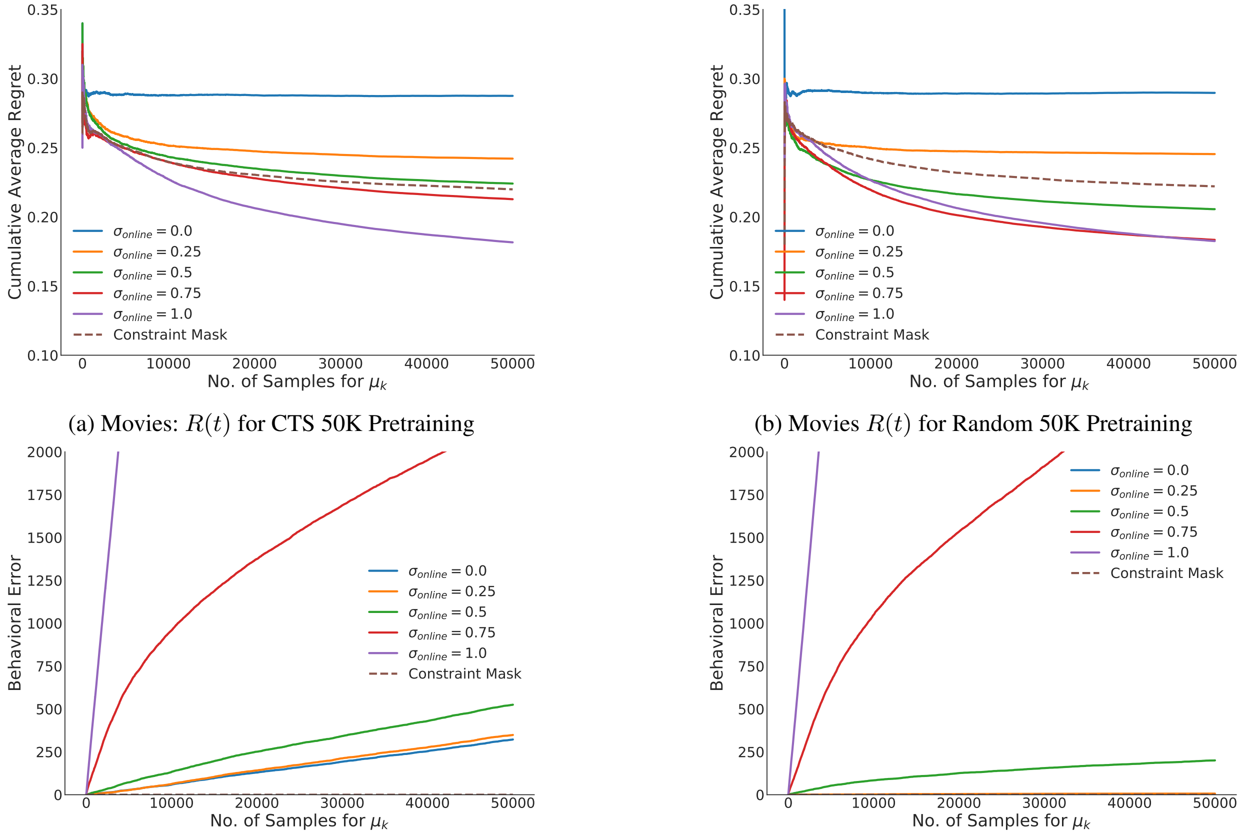
Theorem 1. *Consider the BCTS algorithm with K arms, d features, and take $0 \leq \sigma_{online} \leq 1$ and $0 \leq \gamma \leq 1$. Then, with probability $(1-\gamma)$, the upper bound on the regret R(T) at time T is:*

$$\sigma_{online} \frac{d\gamma}{z} \sqrt{T^{z+1}} (ln(T)d) ln\frac{1}{\gamma} +$$
$$(1 - \sigma_{online}) c_{max} T ||\tilde{\mu}^*_{max} + \mu^e_{min}||_2$$

where c_{max} a positive constant and $0 < z < 1$, a constant parameter of the CTS algorithm, σ_{online} a distance threshold and $\mu^e_{max} = max_k(||\mu^e_k||_1)$, $\tilde{\mu}^*_{max} = max_k(||\tilde{\mu}^*_k||_1)$ with $k \in K$.

The above theorem tells us that, if the constrained policy μ^e and the optimal policy $\tilde{\mu}^*$ are close to each other, then we can recover the bound on CTS given as a Lemma by Agrawal and Goyal (2013). Hence, it is interesting for future work to study distributions of the data and find data driven bounds on the regret.

Definition 2 (Optimal Policy for BCCB)**.** *The optimal policy for solving the BCCB is selecting the arm at time t: $k(t) = \underset{k \in K}{argmax}\ (\sigma^*_{online} \tilde{\mu}^*_k(t) + (1 - \sigma^*_{online}) \tilde{\mu}^{e*}_k(t))^\top c(t)$, where $\tilde{\mu}^*_k$ the optimal mean vector for the reward driven pol-*



(a) Movies: $R(t)$ for CTS 50K Pretraining

(b) Movies $R(t)$ for Random 50K Pretraining

(c) Movies $E(T)$ for CTS 50K Pretraining

(d) Movies $E(T)$ for Random 50K Pretraining

Figure 1: Results for Movie datasets; mean over 5 CVs. Figures (a-b) show the cumulative average regret $R(t)$ as we vary σ_{online} for random and CTS based pretraining; μ^e was trained with 50,000 examples and compared to a Constraint Mask baseline that is explicitly given the constraints. Figures (d-e) shows the behavioral error $E(t)$, i.e., the total number of constraints violated. Increasing the sensitivity to the online reward improves the performance of the agent significantly w.r.t. the cumulative average regret. Notice the scale difference between the axes in (c-d), $\sigma_{online} = 1.0$ is making a linear number of constraint violations while other settings give sublinear numbers of violations.

icy $\tilde{\mu}^{e*}_k$ the optimal mean vector for the behavior constraint driven policy.

Theorem 2. *Using Definition 2 of optimal policy we have, with probability $(1-\gamma)$, the upper bound on the regret R(T) at time T is:*

$$max(\sigma^*_{online}, \sigma_{online})\frac{d\gamma}{z}\sqrt{T^{z+1}}(ln(T)d)ln\frac{1}{\gamma} +$$

$$\max((1-\sigma^*_{online}), (1-\sigma_{online}))\frac{d\gamma}{z}\sqrt{N^{z+1}}(ln(N)d)ln\frac{1}{\gamma}$$

where c_{max} a positive constant and $0 < z < 1$, a constant parameter of the CTS algorithm, σ^*_{online} the optimal distance threshold.

Theorem 2 is interesting in that, if we use Definition 2 as our optimal policy, the regret of our proposed solution is sub-linear in time T and sub-linear in the number of training examples N. We will compare using the CTS algorithm to a random baseline in our experiments in the next section.

5 Experimental Evaluation

To study the effect of imposing exogenous constraints on an online decision making agent and to demonstrate the soundness and flexibility of our techniques, we perform a set of experiments using real world data. As we are unaware of any prior work that we can directly compare with where a constraint is learned, or inferred, and then used to bound or control the behavior of an online agent, we construct three reasonable baselines for comparison. First, an agent who perfectly knows the constraints and is able to completely obey them, second, an agent who is follows a Thompson Sampling approach to learn the constraints, and finally, an agent that learns the constraints through random sampling of the space during the teaching phase.

Movie Data

We build an online movie recommendation agent (Mary, Gaudel, and Preux 2015) that learns online what movies to recommend to protected users (such as kids or older people) but also follows some guidelines set by parents or relatives

	Mask		$\sigma_{online} = 0.0$		$\sigma_{online} = 0.25$		$\sigma_{online} = 0.50$		$\sigma_{online} = 0.75$		$\sigma_{online} = 1.00$	
N	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$
5K	0.214	0	0.310	787.4	0.279	572.0	0.240	878.0	0.204	4834.2	0.176	22006.8
10K	0.216	0	0.274	296.0	0.240	231.8	0.212	423.2	0.191	2325.8	0.178	24706.2
50K	0.220	0	0.288	322.2	0.242	348.6	0.224	525.0	0.212	2198.4	0.182	23390.4
75K	0.213	0	0.316	224.2	0.248	180.0	0.235	308.4	0.227	2278.6	0.178	23611.8
100K	0.207	0	0.284	489.0	0.220	542.4	0.206	734.2	0.201	2412.6	0.171	22900.0

Table 1: Results on Movies varying the number of training examples N and σ_{online} with CTS. $T = 50,000$; mean over 5 CV.

	Mask		$\sigma_{online} = 0.0$		$\sigma_{online} = 0.25$		$\sigma_{online} = 0.50$		$\sigma_{online} = 0.75$		$\sigma_{online} = 1.00$	
N	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$
5K	0.215	0	0.293	177.2	0.275	258.4	0.241	646.2	0.201	3044.8	0.179	24155.2
10K	0.224	0	0.283	32.2	0.263	31.2	0.228	201.8	0.187	1986.6	0.183	27502.8
50K	0.222	0	0.289	1.0	0.245	6.6	0.205	200.2	0.183	2582.4	0.182	27541.0
75K	0.220	0	0.284	0.2	0.229	5.8	0.204	265.8	0.179	2433.8	0.181	27499.0
100K	0.229	0	0.277	0.0	0.222	4.6	0.193	212.0	0.181	2416.6	0.181	27654.2

Table 2: Results on Movies varying N and σ_{online} with random sampling. $T = 50,000$; mean over 5 CV.

over which movies are suitable (and which are not) for their kids (or older members of the family). For example, parents may think that young people should not be recommended movies with too much violence, and that older people should not be recommended horror movies. Reward is given when the user reviews the movie, as in Netflix. We want the online agent to accrue as much reward as possible, while at the same time not violating or straying too far from the constraints. We now describe the data and how we created the constraints in this setting.

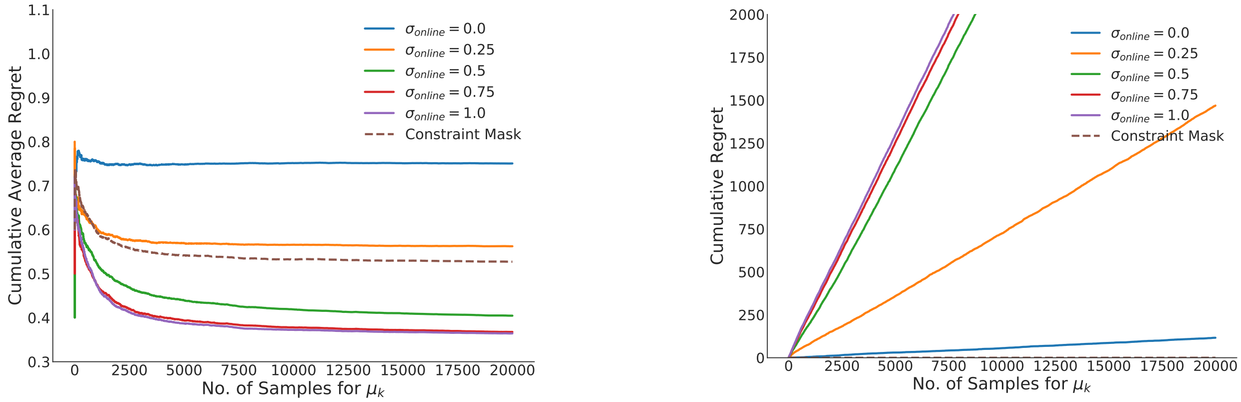
Data. We start from the MovieLens 20m dataset (Harper and Konstan 2016), which contains 20 million ratings of 27,000 movies by 138,000 users along with genre information. We subset this by taking the top 100 users by number of movies rated and top 1000 movies by number of ratings received. The subset of the data from MovieLens is sparse in terms of the ratings of the movies by the users. Since we resample to run many experiments over the same dataset, we create a complete rating matrix with a user based collaborative filtering pass, rounded to increments of 0.5 to match the rest of the data (Bell and Koren 2007; Schafer et al. 2007).

The genre information, which defines the context vector, for movie m is $c^d(m)$ is a $d = 10$ dimensional vector consisting of one or more of the categories: Action, Adventure, Comedy, Drama, Fantasy, Horror, Romance, Sci-Fi, Thriller. We impute ages to the users using the bands that marketers most often used in advertising: 12-17, 18-24, 25-34, 35-44, 45-54, 55-64, 65+ (Jobber and Ellis-Chadwick 2012). This is drawn randomly for each user using the population by age and sex demographics from the US Census.[1]

Methodology. The MovieLens data does not contain any notion of constraints so we must construct the constraints to test our methods. While in the real world the we assume the behavior constraint is not given explicitly we create a *behaviorally constrained training set* derived from an explicit *behavior constraint matrix* to train our agent. The behavior constraint matrix is a $\{0, 1\}$ matrix of size $a \times d = 7 \times 10$ representing whether or not we can recommend a movie with genre type d to someone of age type a. We convert this to a behavioral training set (matrix) of size $users \times movies$ where each cell contains the behavioral reward of recommending movie m to user u. We enforce a constraint in the most restricted sense, i.e., if a movie has multiple features then, if any feature is restricted, the movie is as well. For example, if the behavioral constraint matrix has a 0 in the entry at at the $(12 - 17) \times$ (Horror) location then we are not allowed to suggest any movie m which has $c(m)$(Horror) $= 1$ to any user (arm) u which has $\mathbf{f}(u)$(12-17) $= 1$. To learn the behaviorally constrained policy μ^e during the constraint learning phase we use a disjoint subset of 200 movies (contexts), but the same arms (users), from those used in the recommendation phase. During this phase we use both the classic CTS algorithm as well as random exploration and vary the number queries allowed to learn μ^e. For the recommendation phase we show the contexts (movies) to the agent as we vary σ_{online} between $\sigma_{online} = 0$, always follow μ_e and $\sigma_{online} = 1$, always follow $\tilde{\mu}$. For each setting of σ_{online} we show the movies in the same order so our results are comparable across settings of σ_{online}.

Results. During the online runs we track the cumulative average regret at time T, $R(T) = \sum_{t=1}^{T} r_{\mu^*}(t) - r_{\tilde{\mu}}(t) / T$, where $r_{\mu^*}(t)$ is the reward for the best action the agent could have taken at time t and $r_{\tilde{\mu}}$ is the reward for the action actually taken. We also track the behavioral error at time T which is given by $E(T) = \sum_{t=1}^{T} r^e_{\tilde{\mu}}(t)$. We vary the value of σ_{online} and compare these results to an omniscient agent who knows the constraints exactly and applies them as a mask, hence it never violates the behavioral constraints and is accumulating as much reward as possible under the constraints. The results of these experiments are depicted in Figure 1 as well as Table 1 and 2 where *Constraint Mask* is the agent who is given the constraints explicitly. For each of these experiments, we show the means over 5 cross validations using a

[1] https://factfinder.census.gov/



(a) Warfarin: $R(t)$ for CTS 50K Pretraining

(b) Warfarin $E(t)$ for CTS 50K Pretraining

Figure 2: Results for Warfrin datasets; mean over 5 CVs. Figure (a) shows the cumulative average regret $R(t)$ as we vary σ_{online}, μ^e was trained using CTS and 50,000 examples v. Constraint Mask baseline that is provided the constraints. Figure (b) shows the behavioral error $E(t)$, i.e., the total number of constraints violated. Increasing the sensitivity to the online reward improves the performance of the agent significantly. At $\sigma_{online} = 0.25$ there are a smaller number of behavioral errors and significantly reduce regret, though, due to the nature of the rewards in this domain performance is not as good as in Figure 1

different subset of 200 movies to train μ^e each time. This gives us plots depicted in Figure 1 and the extended results in Tables 1 and 2. From this we can say that the agent is consistent within this setting.

Looking first at Figures 1a and 1b we see the results when we impose a number of constraints on the agent as we set σ_{online} to various values; μ^e was trained with 50,000 samples for all graphs. Recall that when $\sigma_{online} = 0.0$ the agent is not sensitive to the reward and hence has consistent behavior during the online phase, accumulating a larger amount of regret than the other agents but incurring less behavioral error. The interesting result comes when allowing some sensitivity to the online feedback ($\sigma_{online} = 0.50$) as we get, for both the CTS and the random pretraining, drastically better performance in terms of balancing $R(t)$ and $E(t)$ which can be seen by contrasting with Figures 1c and 1d which plot $E(t)$. Encouragingly, looking at Figures 1c and 1d we see that while the masked agent never makes any errors, the $\sigma_{online} = \{0.0, 0.25, 0.50\}$ agents are able to make a very small number of errors, indicating that we have been able to learn the constraints well from only the examples.

Turning to the question of which method of pretraining is better, random or CTS, Tables 1 and 2 give us a closer look. These tables show $R(T)$ and $E(T)$ with $T = 50,000$ for the movie domain when one agent was trained with CTS and the other with random. Surprisingly, we see that the random pretraining agent is *better* at learning the constraints across the board than the agent trained with CTS; making significantly fewer errors when $\sigma_{online}\{0.00, 0.25, 0.50\}$ and decreasing the total error, $E(T)$ as we train it with more examples. Going so far as to behave optimally when we set $\sigma_{online} = 0.0$ and pretrain with 100K examples. Most interestingly, we see in Table 2 that the random agent for $\sigma_{online} = 0.25$ and 100K examples is able to both make no behavioral errors, i.e., $E(T) = 0$ *and* out perform the Constraint Mask in terms of $R(T)$. From this table we can infer two things: (1) that our random agent is able to learn a high quality constrained policy given only examples and (2) that it is able to deploy this policy in the online recommendation phase.

Healthcare Data

We consider a dataset for the healthcare domain, specifically about the drug Warfarin. Warfarin is one of the most used anticoagulants in the world and correct dosing is a standing challenge (Wysowski, Nourjah, and Swartz 2007). Over 40,000 emergency room visits a year are related to Warfarin dosing (Budnitz et al. 2006) and, for this reason, large datasets are available that track upwards of 50 patient factors along with the appropriate dosing levels.

Data & Methods. In this dataset the context dimensionality $d = 39$ and includes factors such as age, sex, and presence of interacting drugs including acetaminophen. The dosing options, the arms here, is in three levels: low, medium, and high. To create behavioral constraints we suppose that there are two additional features, randomly distributed amongst the patients, such that the presence of both of these features would cause a significant decrease in the quality of life of the patient. We add a *no dose* arm, and enforce, as a behavioral constraint, that patients should not be prescribed Warfarin if both additional features are present. Reward in the online case is obtained by the agent for prescribing the correct dosing level, hence there will always be a cost associated with following the behavioral constraints in this setting. The rest of our methodology is unchanged from the movie dataset.

Results. We track the same statistics as for the movie domain. The results are depicted in Figure 2 when we train μ_e with 50,000 teaching examples. First, we see that our agent is able to learn a very good constrained policy in this space and follow it. Again, Our $\sigma_{online} = 0.25$ agent with 50,000 examples is able to have regret on par with the Constraint Mask agent while only making 1,500 mistakes over 20,000 trials versus nearly 20,000 for all the other agents.

	Mask		$\sigma_{online} = 0.0$		$\sigma_{online} = 0.25$		$\sigma_{online} = 0.50$		$\sigma_{online} = 0.75$		$\sigma_{online} = 1.00$	
N	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$
5K	0.525	0	0.737	178.2	0.558	1488.0	0.386	4828.0	0.367	5005.6	0.369	5106.4
10K	0.526	0	0.732	73.2	0.594	765.4	0.386	4793.0	0.366	4980.6	0.365	5090.4
50K	0.528	0	0.759	92.8	0.522	2704.8	0.380	4940.8	0.368	5049.4	0.368	5141.6
75K	0.527	0	0.739	88.2	0.390	4752.2	0.371	4918.2	0.369	4977.4	0.371	5059.2
100K	0.525	0	0.737	85.6	0.383	4713.6	0.367	4943.0	0.364	5013.2	0.365	5091.2

Table 3: Results on Warfarin varying the number of training examples N and σ_{online} with CTS. $T = 20,000$; mean over 5 CV.

	Mask		$\sigma_{online} = 0.0$		$\sigma_{online} = 0.25$		$\sigma_{online} = 0.50$		$\sigma_{online} = 0.75$		$\sigma_{online} = 1.00$	
N	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$	$R(T)$	$E(T)$
5K	0.526	0	0.765	91.0	0.553	210.8	0.387	4344.4	0.366	5022.8	0.364	5133.6
10K	0.527	0	0.786	99.8	0.541	180.0	0.497	3818.4	0.372	4915.2	0.370	5033.2
50K	0.527	0	0.732	135.8	0.523	182.8	0.393	4173.4	0.368	4988.8	0.370	5110.8
75K	0.528	0	0.764	126.6	0.532	164.0	0.399	4102.8	0.374	4960.6	0.372	5074.4
100K	0.526	0	0.755	143.0	0.525	169.8	0.392	4151.4	0.367	4997.2	0.367	5114.0

Table 4: Results on Warfarin varying N and σ_{online} with random sampling. $T = 20,000$; mean over 5 CV.

Allowing increasing the sensitivity to the online reward improves the performance of the agent but, in this setting, the constraints are not orthogonal to the rewards, instead the overlap. Thus the only way to collect additional reward is to violate the constraints and we see this in the high number of errors necessary in order to decrease regret. However, the $\sigma_{online} = 0.25$ agent is able to perform on par with the Constraint Mask agent with only about 75 errors per 1,000 pulls. The high dimensionality the Warfarin data can cause problems for convergence when modeled as CMAB problem (Bastani and Bayati 2015). However, we we still achieve good results for the constrained behavior in this setting. Looking at Tables 3 and 4 we see different results from the Movies domain: both CTS and Random are able to learn the constraints well with a very small edge for CTS here when $\sigma_{online} = 0.0$ and a larger number of examples.

Discussion

Looking at both experimental domains, we see that using either random or CTS we are able to learn a high quality constrained policy described only by examples of appropriate behavior. While we have theoretical bounds for the quality of our CTS agent, we see that the random agent is able to outperform it in practice. Regardless, the agent is able to use this policy to guide the actions it takes during the recommendation phase. This agent is able to make decisions that very rarely (or never) violate the constrained policy, achieving performance on par with an agent that is given the behavioral constraints explicitly. Understanding the specific interaction between the distribution of the behavioral constraints and the rewards in the data is an important direction for future work. We saw that in some cases we are able to follow the constraints without affecting the accumulated rewards, while in other cases it had a large impact on the quality of the agents decisions. We also plan to use an active learning approach (Krishnamurthy et al. 2017) to the problem, where the online agent is presented the option to query the omniscient constraint agent during the online phase.

6 Conclusions

In real life scenarios agents that recommend items to humans are subject to a plethora of exogenous priorities, given by ethical principles, moral values, social norms, and professional codes. It is essential that AI system are able to understand decisions based on their compliance to such constraints in order to reach suitable tradeoffs between satisfying user's desires and behaving appropriately. We propose to achieve this tradeoff by extending the CMAB problem machinery to include exogenous constraints, modeled in the form of a policy learned during a constraint learning phase from observing a teaching agent. Our evaluation over real data shows that our system can act within these behavior constraints while not significantly degrading the reward. An important direction for future work is understanding the interaction between the behavioral constraints and the online rewards for various data distributions as well as extending to domains where we have sequences of actions (Noothigattu et al. 2018).

References

Abel, D.; MacGlashan, J.; and Littman, M. L. 2016. Reinforcement learning as a framework for ethical decision making. In *Workshops of the 30th AAAI: AI, Ethics, and Society*, 54–61.

Agrawal, S., and Goyal, N. 2012. Analysis of Thompson sampling for the multi-armed bandit problem. In *Proc. 25th COLT*, 39.1–39.26.

Agrawal, S., and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *ICML (3)*, 127–135.

Agrawal, S., and Goyal, N. 2016. Linear contextual bandits with knapsacks. In *Proc. NIPS 2016*, 3450–3458.

Allen, C.; Smit, I.; and Wallach, W. 2005. Artificial morality: Top-down, bottom-up, and hybrid approaches. *Ethics and Information Technology* 7(3):149–155.

Anderson, M., and Anderson, S. L. 2011. *Machine Ethics*. Cambridge University Press.

Armstrong, S. 2015. Motivated value selection for artificial agents. In *Workshops of the 29th AAAI: AI, Ethics, and Society*.

Auer, P., and Cesa-Bianchi, N. 1998. On-line learning with malicious noise and the closure algorithm. *Ann. Math. Artif. Intell.* 23(1-2):83–99.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2002. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.* 32(1):48–77.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47(2-3):235–256.

Balakrishnan, A.; Bouneffouf, D.; Mattei, N.; and Rossi, F. 2018. Using contextual bandits with behavioral constraints for constrained online movie recommendation. In *Proc. 27th IJCAI*.

Bastani, H., and Bayati, M. 2015. Online decision-making with high-dimensional covariates. Technical report, SSRN.

Bell, R. M., and Koren, Y. 2007. Lessons from the Netflix prize challenge. *SIGKDD Explorations* 9(2):75–79.

Bonnefon, J.-F.; Shariff, A.; and Rahwan, I. 2016. The social dilemma of autonomous vehicles. *Science* 352(6293):1573–1576.

Bouneffouf, D.; Rish, I.; Cecchi, G. A.; and Féraud, R. 2017. Context attentive bandits: Contextual bandit with restricted context. In *Proc. 26th IJCAI*, 1468–1475.

Briggs, G., and Scheutz, M. 2015. "Sorry, I can't do that": Developing mechanisms to appropriately reject directives in human-robot interactions. In *AAAI Fall Symposium Series: AI for Human-Robot Interaction*.

Budnitz, D. S.; Pollock, D. A.; Weidenbach, K. N.; Mendelsohn, A. B.; Schroeder, T. J.; and Annest, J. L. 2006. National surveillance of emergency department visits for outpatient adverse drug events. *Journal of the American Medical Association* 296(15):1858–1866.

Chu, W.; Li, L.; Reyzin, L.; and Schapire, R. E. 2011. Contextual bandits with linear payoff functions. In *Proc. AISTATS*, 208–214.

Harper, F. M., and Konstan, J. A. 2016. The Movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5(4):19.

Jobber, D., and Ellis-Chadwick, F. 2012. *Principles and Practice of Marketing*. McGraw-Hill Higher Education.

Kaufmann, E.; Korda, N.; and Munos, R. 2012. Thompson Sampling: An Asymptotically Optimal Finite Time Analysis. In *Proc. 23rd Algorithmic Learning Theory (ALT)*, 199–213.

Krishnamurthy, A.; Agarwal, A.; Huang, T.; III, H. D.; and Langford, J. 2017. Active learning for cost-sensitive classification. In *Proc. of the 34th ICML*, 1915–1924.

Kveton, B.; Wen, Z.; Ashkan, A.; Eydgahi, H.; and Eriksson, B. 2014. Matroid bandits: Fast combinatorial optimization with learning. In *Proc. of the 30th UAI*, 420–429.

Lai, T. A., and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics* 6(1):4–22.

Langford, J., and Zhang, T. 2008. The Epoch-Greedy Algorithm for Contextual Multi-armed Bandits. In *Proc. 21st NIPS*.

Leike, J.; Martic, M.; Krakovna, V.; Ortega, P.; Everitt, T.; Lefrancq, A.; Orseau, L.; and Legg, S. 2017. AI safety gridworlds. *arXiv preprint arXiv:1711.09883*.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proc. 19th WWW*, 661–670.

Loreggia, A.; Mattei, N.; Rossi, F.; and Venable, K. B. 2018a. Preferences and ethical principles in decision making. In *Proc. 1st AAAI/ACM AIES*.

Loreggia, A.; Mattei, N.; Rossi, F.; and Venable, K. B. 2018b. Value alignment via tractable preference distance. In Yampolskiy, R. V., ed., *Artificial Intelligence Safety and Security*. CRC Press. chapter 16.

Mary, J.; Gaudel, R.; and Preux, P. 2015. Bandits and recommender systems. In *Machine Learning, Optimization, and Big Data*, 325–336.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing Atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Noothigattu, R.; Bouneffouf, D.; Mattei, N.; Chandra, R.; Madan, P.; Varshney, K.; Campbell, M.; Singh, M.; and Rossi, F. 2018. Interpretable multi-objective reinforcement learning through policy orchestration. *arXiv preprint arXiv:1809.08343*.

Rossi, F., and Mattei, N. 2019. Building ethically bounded AI. In *Proc. 33rd AAAI (Blue Sky Track)*.

Russell, S.; Dewey, D.; and Tegmark, M. 2015. Research priorities for robust and beneficial artificial intelligence. *AI Magazine* 36(4):105–114.

Schafer, J. B.; Frankowski, D.; Herlocker, J. L.; and Sen, S. 2007. Collaborative filtering recommender systems. In *The Adaptive Web, Methods and Strategies of Web Personalization*, 291–324.

Sen, A. 1974. Choice, ordering and morality. In Körner, S., ed., *Practical Reason*. Oxford: Blackwell.

Shivaswamy, P. K., and Joachims, T. 2012. Multi-armed bandit problems with history. In *Proc. of the 15th AISTATS*, 1046–1054.

Sutton, R. S., and Barto, A. 2017. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition.

Thompson, W. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25:285–294.

Villar, S. S.; Bowden, J.; and Wason, J. 2015. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical Science* 30(2):199.

Wallach, W., and Allen, C. 2008. *Moral machines: Teaching robots right from wrong*. Oxford University Press.

Wu, H.; Srikant, R.; Liu, X.; and Jiang, C. 2015. Algorithms with logarithmic or sublinear regret for constrained contextual bandits. In *Proc. NIPS*, 433–441.

Wu, Y.; Shariff, R.; Lattimore, T.; and Szepesvári, C. 2016. Conservative bandits. In *Proc. of 33rd ICML*, 1254–1262.

Wysowski, D. K.; Nourjah, P.; and Swartz, L. 2007. Bleeding complications With warfarin use: a prevalent adverse effect resulting in regulatory action. *Archives of Internal Medicine* 167(13):1414–1419.

Yu, H.; Shen, Z.; Miao, C.; Leung, C.; Lesser, V. R.; and Yang, Q. 2018. Building ethics into artificial intelligence. In *Proc. 27th IJCAI*, 5527–5533.