From Ads to Interventions: Contextual Bandits in Mobile Health

Ambuj Tewari and Susan A. Murphy

Abstract The first paper on contextual bandits was written by Michael Woodroofe in 1979 [1] but the term "contextual bandits" was invented only recently in 2008 by Langford and Zhang [2]. Woodroofe's motivating application was clinical trials whereas modern interest in this problem was driven to a great extent by problems on the internet, such as online ad and online news article placement. We have now come full circle because contextual bandits provide a natural framework for sequential decision making in mobile health. We will survey the contextual bandits literature with a focus on modifications needed to adapt existing approaches to the mobile health setting. We discuss specific challenges in this direction such as: good initialization of the learning algorithm, finding interpretable policies, assessing usefulness of tailoring variables, computational considerations, robustness to failure of assumptions, and dealing with variables that are costly to acquire or missing.

1 Introduction

The classic multi-armed bandit problem (see, e.g., [3]) is perhaps the simplest model of a sequential decision making problem where one wishes to maximize the cumulative sum of rewards received over some time horizon. Faced with a finite number of alternatives, called actions or arms, the decision maker must choose between them at every time point. One has to balance the exploration of actions that have hitherto yielded low rewards, with exploitation of current knowledge about actions have yielded high rewards so far.

Woodroofe [1] noted that, in most sequential decision making scenarios, there is likely to be some additional information available that can be useful for decision

Ambuj Tewari
University of Michigan, Ann Arbor, e-mail: tewaria@umich.edu

Susan A. Murphy
University of Michigan, Ann Arbor, e-mail: samurphy@umich.edu

making. For example, in a clinical trial with two drugs, we might have people's genetic or demographic information available as features. If so, then rather than thinking about a two-armed bandit problem, one should think about the clinical trial as a *contextual bandit* problem where we want to learn how to map user features into one of the available actions, i.e., one of the two drugs in this case. Woodroofe defined this problem, albeit in the case of just one feature, but he did not call it a "contextual bandit" problem. Instead he called it a "bandit problem with a concomitant variable".

As it sometimes the case with broadly useful problems, contextual bandit problems have been considered by many different communities by many different names. They have been called "bandit problems with side observations" [4, 5], "bandit problems with side information" [6], "associative reinforcement learning" [7, 8, 9], "reinforcement learning with immediate reward" [10], "associative bandit problems" [11], and "bandit problems with covariates" [12, 13, 14, 15]. The term "contextual bandits" was coined by Langford and Zhang [2] and we stick to it because it is descriptive yet short.

Recent interest in contextual bandits has been driven to a large extent by personalization problems arising on the web. How to use user and webpage features to select the best ad to show to the user on a given webpage [16]? How to show personalized news articles to web users based on their interests [17]? With the emergence of mobile health, we expect that many ideas developed to show personalized ads to users on the web will be found useful in personalizing mobile health interventions to a specific person in a particular context.

The framework of Just-In-Time Adaptive Interventions [18] has recently been put forward to unify a number of decision making problems that arise in mobile health across a variety of behavior change domains including alcohol abuse, depression, obesity, and substance abuse. There are five keys components of JITAIs: decision points, decision rules, tailoring variables, intervention options, and proximal outcomes. Contextual bandit algorithms can be used for personalizing JITAIs. The tailoring variables, such as GPS location, calendar busyness, and heartrate, form the context. The intervention options are the actions. For simplicity, we assume throughout this chapter, that there are only two intervention options: whether to intervene or not. For example, in a physical activity JITAI, the two intervention options might be whether or not to send an activity encouraging message. Once an intervention option is chosen, a proximal outcome (i.e., reward) is obtained. Again, to use the example of the physical activity JITAI, our proximal outcome might be the number of steps the person walked in the one hour following the decision point. In JITAIs, the fundamental pattern that repeats over time is the following.

1: **at** a given decision point **do**
2: mobile phone collects tailoring variables (the context)
3: a decision rule (or policy) maps the tailoring variables into an intervention option (the action)
4: mobile phone records the proximal outcome (interpreted as a reward, so higher is better)
5: **done**

In the rest of this chapter, we will see how the contextual bandit problem is a good way to think about the problem of personalizing JITAIs in a mobile health setting. We will look at online learning algorithms that learn good decision rules (policies) over time by interacting with the environment using a protocol very similar to the fundamental temporally-repeating pattern described above. We will first survey existing contextual bandit frameworks and algorithms to give the reader a sense of the breadth of work that has occurred in this area across several different fields including computer science, electrical engineering, operations research, and statistics. Then we will highlight the unique challenges that arise in mobile health and discuss how existing contextual bandit algorithms will need to be modified before they can be used successfully in mobile health.

2 Online Learning in Contextual Bandits

In this section we will review the online learning literature on contextual bandit problems. The focus will be on algorithms that minimize their *regret*. Regret measures the difference between the reward that could have been accumulated with prior knowledge of the problem, and the reward accumulated by the learning algorithm. The precise definition depends on the setting in which one is analyzing the learning algorithm. We will consider three settings that make increasingly weaker assumptions about the data generating process. In the first setting, contexts and rewards are all stochastically generated from an iid process. In the second setting, contexts are arbitrary but rewards are stochastic. Finally, in the third setting, contexts and rewards are all arbitrary.

2.1 Stochastic Contextual Bandits

In the stochastic setting, we assume that the context and reward triples $\{(X_t, R_t^0, R_t^1)\}_{t=1}^T$ are generated by sampling independently from an underlying distribution $\mathscr{D}$. The following online learning protocol is followed.

1: **for** $t = 1$ to T **do**
2: receive context X_t
3: algorithm takes action A_t
4: receive reward $R_t = R_t^{A_t}$
5: **end for**

The contexts X_t are drawn from some context space $\mathscr{X}$. Unless otherwise specified, we will assume that the context $X_t \in \mathbb{R}^p$ is a vector with p components so that $\mathscr{X} \subseteq \mathbb{R}^p$. The literature has considered situations both less general (e.g., finite context space [19]) and more general (e.g., contexts in a general metric space [20]). The actions A_t lie an action space $\mathscr{A}$ which we will assume, unless indicated otherwise,

to be $\{0,1\}$ with 1 corresponding to the option of providing an intervention and 0 to not providing.

A *policy* or *decision rule* $\pi : \mathscr{X} \to \mathscr{A}$ decides which action gets taken in which contexts. The *value* of a policy π is defined as the expected reward obtained when actions are chosen according to π:

$$V(\pi) = \mathbb{E}_{(X,R^0,R^1)\sim\mathscr{D}}\left[R^{\pi(X)}\right].$$

The value of a policy, in turn, depends on the expected reward functions $\eta_a, a \in \mathscr{A}$, defined as:

$$\eta_a(x) = \mathbb{E}_{(X,R^0,R^1)\sim\mathscr{D}}[R^a|X = x].$$

Note that the value of a policy and the expected reward functions are related to each other as follows:

$$V(\pi) = \mathbb{E}_{X\sim\mathscr{D}_X}\left[\eta_{\pi(X)}(X)\right].$$

Here $\mathscr{D}_X$ is the marginal distribution of contexts. The optimal policy $\pi^\star$, among all possible policies, is given by

$$\pi^\star(x) = \operatorname*{argmax}_{a\in\mathscr{A}} \eta_a(x). \tag{1}$$

An *online learning algorithm* $\mathscr{L}$ is a sequence of maps $\mathscr{L}_t, 1 \le t \le T$, where $\mathscr{L}_t$ maps the history just prior to time t, $\{(X_s,A_s,R_s)\}_{s=1}^{t-1}$, along with the current context X_t to an action $A_t \in \mathscr{A}$. If any of the maps $\mathscr{L}_t$ are stochastic, i.e., the algorithm uses some internal randomization, then we call it a *randomized online learning algorithm*. Otherwise, we call it a *deterministic online learning algorithm*.

We will look at several different notions of *regret*. All of them will be of the form:

$$\text{“best expected cumulative reward in a comparison class”} - \sum_{t=1}^{T} \mathbb{E}[R_t]$$

where the first term, referred to as the “benchmark” or “comparator” term measures the total expected reward that would have been obtained with advanced knowledge of the distributions (in the stochastic case) or nature's moves (in the adversarial case). The second term is the expected reward accumulated by the online learning algorithm. Note that this expectation is taken with respect to any randomness in nature's generation of contexts and rewards, as well as any randomness used by the algorithm (if it is a randomized online learning algorithm).

Contextual bandit problems can be approached through several perspectives. We can adopt a *regression* perspective and view the problem as one of estimating the expected reward functions $\eta_a(x)$. Given estimates $\widehat{\eta}_a$, we can choose the corresponding “greedy” policy $\textsc{Greedy}(\widehat{\eta}_a)$ defined as

$$\textsc{Greedy}(\widehat{\eta}_a)(x) = \operatorname*{argmax}_{a\in\mathscr{A}} \widehat{\eta}_a(x).$$

Note that the optimal policy defined in (1) is nothing but $\textsc{Greedy}(\eta_a)$.

In the case of two actions, one can also adopt a *binary classification* perspective and fix a set Π of policies that can also be thought of as a set of classifiers. The best policy in this class is

$$\pi^\star_\Pi = \operatorname*{argmax}_{\pi\in\Pi} V(\pi).$$

Instead of estimating the underlying expected reward function, one can instead simply try to compete with $\pi^\star_\Pi$.

In the rest of this section, we first review approaches, both parametric and non-parametric, based on the regression perspective. Then we will consider classification based approaches that search for good policies in a restricted class.

2.1.1 Parametric Estimation of Expected Reward Functions

In addition to assuming that the triples (X_t, R^0_t, R^1_t) are iid, let us also assume that

$$R^a_t = \beta_a^\top X_t + \varepsilon^a_t, \tag{2}$$

where $X_t, \beta_a^\top \in \mathbb{R}^p$ and ε^a_t are iid mean-zero random variables. This implies that the expected reward functions $\eta_a(x) = \beta_a^\top x$ are linear in the context x. Under this assumption, the best policy takes the form

$$\pi^\star(x) = \textsc{Greedy}(\beta_a)(x) = \operatorname*{argmax}_{a\in\mathscr{A}} \beta_a^\top x.$$

Expected reward of this optimal policy over T time steps is $T\cdot V(\pi^\star)$. The expected regret of a learning algorithm is defined as

$$T\cdot V(\pi^\star) - \sum_{t=1}^{T} \mathbb{E}[R_t]. \tag{3}$$

A simple approach for online learning in this setting is to adopt what has been called a "certainty equivalence with forcing" strategy in the adaptive control literature [21]. The idea is to choose a predetermined sequence of time points when the learning algorithm simply explores different actions. On rounds other than the exploration rounds, the algorithm "exploits" the current knowledge. "Greedy" or "certainty equivalent" exploitation means that the algorithm believes its current estimates of the expected reward function and takes the optimal action according to those estimates.

The algorithm of Goldenshluger and Zeevi [22] (Algorithm 1) adopts such an approach with a slight twist: it maintains two sets of estimates for the expected reward functions. The first set of estimates, $\widetilde{\beta}_a$, are computed from data obtained during forced exploration rounds and the second set of estimates, $\widehat{\beta}_a$, are computed from data obtained in all previous rounds. At an exploitation round, the algorithm checks to see if there is enough gap between the quality of the two actions according

Algorithm 1 Linear Response Bandit Algorithm [22]

Inputs: n_0 (initial exploration length), $\mathscr{T}_a$ (exploration times for action a), h (localization parameter to decide which estimates to use)

for $t = 1$ to $2n_0$ **do**
  Take action $A_t = 0$ or $A_t = 1$ depending on whether t is odd or even
end for
for $t = 2n_0 + 1$ to T **do**
  if $t \in \mathscr{T}_a$ **then**
    $/\star$ Exploration round $\star/$
    Take action $A_t = a$
    Update $\widehat{\beta}_a$ using least squares on previous rounds when action a was taken
    Update $\widetilde{\beta}_a$ using least squares on previous exploration rounds when action a was taken
  else
    $/\star$ Exploitation round $\star/$
    if $|(\widetilde{\beta}_1 - \widetilde{\beta}_0)^\top X_t| > h/2$ **then**
      Take action $A_t = \text{argmax}_a (\widetilde{\beta}^a)^\top X_t$
    else
      Take action $A_t = \text{argmax}_a (\widehat{\beta}^a)^\top X_t$
    end if
  end if
end for

to $\widetilde{\beta}_a$. If there is enough gap, then it selects an action using the policy $\textsc{Greedy}(\widetilde{\beta}_a)$, otherwise it uses the policy $\textsc{Greedy}(\widehat{\beta}_a)$.

Goldenshluger and Zeevi establish an $O(p^3 \log T)$ regret bound for Algorithm 1 under several assumptions including the assumption that ε_t^a are normally distributed and that a "margin" condition holds. Goldenshluger and Zeevi had earlier brought the margin condition from the classification literature into the contextual bandit literature [23]. The margin condition ensures that the contexts X_t are distributed such that, with high probability, the treatment effect magnitude $|(\beta_1 - \beta_0)^\top X_t|$ is large enough. A margin assumption is problematic in a mobile health setting where treatment effects are often expected to be small.

Recently, Bastani and Bayati [24] have extended Algorithm 1 to the high dimensional case where the vectors β_a are sparse, i.e., the number $\|\beta_a\|_0$ of non-zero elements in β_a satisfies $\|\beta_a\|_0 = s \ll p$. They improve the $O(p^3 \log T)$ regret rate to $O(s^2 \log^2 T + s^2 \log T \log p)$ after making assumptions similar to those made by Goldenshluger and Zeevi.

Linearity of the expected reward function is not the only case that been considered for modeling the expected reward. Agarwal et al. [25] consider a setting where the expected reward function is assumed to lie in a general class with finitely many members. However, extending their results to general, *finite dimensional*, expected reward function classes is an open problem.

2.1.2 Nonparametric Estimation of Expected Reward Functions

Instead of assuming the linear model (2), we can consider the model

$$R_t^a = f_a(X_t) + \varepsilon_t^a, \tag{4}$$

where f_a are functions chosen from a non-parametric class of functions, say, those satisfying certain smoothness conditions, and ε_t^a are iid mean-zero random variables. Assume that the contexts are normalized such that $X_t \in [0,1]^p$.

Algorithm 2 Randomized Allocation with Nonparametric Estimation [14]

Inputs: n_0 (initial exploration length), NPR (nonparametric regression procedure such as nearest neighbor regression), ε_t (sequence of exploration probabilities)

for $t = 1$ to $2n_0$ **do**
  Take action $A_t = 0$ or $A_t = 1$ depending on whether t is odd or even
end for
Get initial estimates $\widehat{f}^a$ by feeding data from previous rounds to NPR
for $t = 2n_0 + 1$ to T **do**
  Let $G_t = \text{argmax}_a \widehat{f}^a(X_t)$ // greedy action
  Let $E_t =$ action selected at random // random exploration
  With probability $(1 - \varepsilon_t)$ take action $A_t = G_t$, else $A_t = E_t$ // ε-greedy
  Collect reward R_t and feed into NPR to get updated estimate $\widehat{f}^a$ for $a = A_t$
end for

Yang and Zhu [14] initiated the study of contextual bandits in this non-parametric setting and looked at the "competitive ratio":

$$\frac{\sum_{t=1}^T f_{A_t}(X_t)}{\sum_{t=1}^T \max_{a \in \mathscr{A}} f_a(X_t)}.$$

Their algorithm, given as Algorithm 2, estimates the functions f_a using some non-parametric procedure such as the histrogram method or the nearest neighbor method. It selects actions using the so-called ε-greedy strategy. That is, with some small probability a random action is selected. Otherwise, the action that looks best according to the current estimates $\widehat{f}_a$ is taken.

Assuming that f_a is non-negative and continuous on $[0,1]^p$ and that $\mathscr{D}_X$ has a density bounded away from zero, Yang and Zhu show that the competitive ratio of their contextual bandit algorithm converges to 1 almost surely, for both the histogram and nearest neighbor methods provided that the width of histograms and number of nearest neighbors are chosen in an appropriate manner as $T \to \infty$.

The results of Yang and Zhu are asymptotic and assume only continuity of the function f_a. Assuming a smoothness condition of the form

$$\forall x, x', a, \ |f^a(x) - f^a(x')| \leq L \cdot \|x - x'\|^\beta,$$

Rigollet and Zeevi [15] gave finite sample expected regret bounds where the expected regret is still defined as in (3) except that now

$$\pi^\star(x) = \textsc{Greedy}(f_a)(x) = \operatorname*{argmax}_{a \in \mathscr{A}} f_a(x).$$

They also assumed a margin condition that controls the probability of observing a context where the treatment effect is non-zero but too small: there exists $\delta_0 \in (0,1)$ such that

$$\forall \delta \in [0, \delta_0), \exists C_\delta \text{ s.t. } \mathbb{P}_{X \sim \mathscr{D}_X}[0 < |f_0(X) - f_1(X)| < \delta] \leq C_\delta \delta^\alpha.$$

If $\alpha\beta > 1$ then the optimal policy $\pi^\star$ does not depend on x and always pulls the same arm. Therefore, to ensure a non-trivial optimal policy, they assume that $\alpha\beta \leq 1$. Their expected regret guarantees are polynomial in T where the exponent depends on the dimension p of the contexts, the margin parameter α and the smoothness parameter β. They also provide almost matching lower bounds. Note these polynomial in T regret rates are much worse than the logarithmic rates in T achievable in the parametric case under margin assumptions.

Perchet and Rigollet [26] extend the work of Rigollet and Zeevi to the case when the number of arms might be (much) larger than 2. They also extend the range of the margin parameter where the bounds hold and eliminate logarithmic gaps between upper and lower bounds. However, their algorithm requires knowledge of the smoothness parameter β. In practice, the smoothness parameter is not known. Qian and Yang [27] show how to use "Lepski-type" procedures from the non-parametric function estimation literature to select the smoothness parameter β in a data-dependent way and still achieve (near) minimax regret bounds that would be obtained assuming that the smoothness is known in advance.

2.1.3 Competing against a Policy Class

In this section, we consider approaches that dispense entirely with the task of estimating the expected reward function. Instead they fix a class Π of policies and aim to minimize the expected regret relative to the class Π, which is defined as

$$T \cdot V(\pi^\star_\Pi) - \sum_{t=1}^{T} \mathbb{E}[R_t], \tag{5}$$

where $\pi^\star_\Pi$ is the best policy in Π.

If the policy class Π is finite ($|\Pi| < \infty$) and small enough that one enumerate all the policies at every time step, then the Exp4 algorithm, given later in Section 2.3.2, can be used. With two actions, it enjoys an expected regret bound of $O(\sqrt{T \log |\Pi|})$ in the fully adversarial setting where the context and reward triples are assumed to be completely arbitrary. If an algorithm enjoys an regret bound in the adversarial setting, it can be shown that it will also satisfy the same bound when the stochastic

Algorithm 3 Epoch Greedy Algorithm [2]

Inputs: Function $\ell(D)$ that given a data set D, outputs the number of exploitation rounds to do next

$D_0 = \{\}, t_1 = 1$
for Epoch $j = 1, 2, \ldots$ **do**
  $t = t_j$
  /⋆ Single exploration step ⋆/
  Select A_t uniformly at random from $\mathscr{A}$
  $D_j = D_{j-1} \cup \{(X_t, A_t, R_t)\}$

  /⋆ Update policy ⋆/
  Compute $\widehat{\pi}_j = \text{argmax}_{\pi \in \Pi} \sum_{(x,a,r) \in D_j} r\mathbf{1}[\pi(x) = a]$

  /⋆ Exploitation phase ⋆/
  $t_{j+1} = t_j + s(D_j) + 1$
  for $t = t_j + 1$ to $t_{j+1} - 1$ **do**
    Take action $A_t = \widehat{\pi}_j(X_t)$
  end for
end for

setting, i.e., when the contexts and rewards are generated by an iid process and regret is measured as in (5) above.

If the policy class is huge or infinite, then enumeration of all policies is infeasible and Exp4 cannot be applied. However, in the stochastic setting, one can use the "certainty equivalence with forcing" idea described in Section 2.1.1 above. Langford and Zhang's [2] Epoch-Greedy algorithm (Algorithm 3) does just that. On an exploration round, it takes one of the two actions at random with probability $1/2$. After an exploration round, it builds an unbiased estimator of the value of any policy π as:

$$\widehat{V}(\pi|D) = \frac{1}{|D|} \sum_{(x,a,r) \in D} 2r\mathbf{1}[\pi(x) = a]$$

where D is the dataset consisting of context, action, reward triples from exploration rounds so far. Since each action is selected at random with probability $1/2$ on exploration rounds, it is easy to see that $\mathbb{E}\left[\widehat{V}(\pi|D)\right] = V(\pi)$ where the expectation is taken over the distribution of contexts and rewards as well as with respect to the algorithm's uniform randomization to select the actions on exploration rounds. The policy selected for the next exploitation phase is then simply

$$\underset{\pi \in \Pi}{\text{argmax}}\ \widehat{V}(\pi|D) = \underset{\pi \in \Pi}{\text{argmax}} \sum_{(x,a,r) \in D} r\mathbf{1}[\pi(x) = a]. \tag{6}$$

This is where the computational advantage of Epoch-Greedy comes in. It never accesses the policies in Π except via the operation above. All we need is a computational blackbox or "oracle" that can answer the "argmax" queries above. Let us call such an oracle an AMO (for Arg Max Oracle). If a cost-sensitive classifier

implementation exists for the class Π then it can serve as an AMO. Therefore, Π can even be infinite as long as an efficient AMO is available for it. The regret bound of Epoch-Greedy, with a finite class Π, is $O(T^{2/3}(\log|\Pi|)^{1/3})$. This is obtained by having $O(T^{2/3}(\log|\Pi|)^{1/3})$ epochs till time T resulting in the same number of AMO calls since exactly one AMO call is made per epoch. Langford and Zhang note that Π need not be finite and that a similar regret bound can be shown for an infinite class with finite VC (Vapnik-Chervonenkis) dimension. Note that for such policy classes, the regret bound of any algorithm that depends on the cardinality of the policy class (such as the one for the Exp4 algorithm in Section 2.3.2 below) becomes vacuous.

Epoch-Greedy's regret guarantee of $O(T^{2/3}(\log|\Pi|)^{1/3})$ might appear to be much worse than logarithmic regret guarantees presented in Section 2.1.1 above. Recall that those guarantees were under additional assumptions such as margin conditions and the constants hidden in the $O(\cdot)$ notation depend on distribution dependent parameters such as the margin parameter. Logarithmic regret guarantees for Epoch-Greedy are possible if one is willing to make additional assumptions and allow distribution dependent constants to appear in the regret guarantee. For instance, consider a finite policy class Π such that there is a *unique* maximizer $\pi^\star$ of the value $V(\pi)$ over π in Π. Let $\Delta > 0$ denote the gap between the value of $\pi^\star$ and that of the second-best policy:

$$\Delta = V(\pi^\star) - \max_{\pi \neq \pi^\star} V(\pi).$$

Langford and Zhang show that Epoch-Greedy also enjoys a regret bound of $O((\log|\Pi| + \log T)/\Delta^2)$. Note that this bound is logarithmic in T but blows up as $\Delta \to 0$.

Dudik et al. [28] gave an algorithm called RandomizedUCB that achieves

$$O(\sqrt{T\log(T|\Pi|/\delta)} + \log(|\Pi|/\delta))$$

regret with probability at least $1-\delta$. Moreover, it requires only polynomially many calls to the AMO at every round. However, its practical utility is still limited as the polynomial involved is of moderately high degree (it invokes the AMO $\tilde{O}(T^5)$ times per round where $\tilde{O}$ hides logarithmic factors). More recent work of Agarwal et al. [29] has managed to bring down the total number of AMO calls to just $O(\sqrt{T/\log(|\Pi|/\delta)})$ *over all T rounds*, with probability at least $1-\delta$, while still preserving the regret bound of RandomizedUCB.

The bandit algorithms discussed above appear quite attractive for use in mobile health due to the fast rate at which the regret decreases to 0. That is, user aggravation and disruption due to inappropriately timed delivery of intervention options would be minimized due to the fast rate at which the algorithm learns the best action for a given context. This is a critical point due to the high levels of app abandonment present in mobile health [30]. However these algorithms achieve these high learning rates under the assumption that the contexts and rewards are all generated from an iid process. Suppose the context includes the user's stress level; user stress at different time points are clearly not independent. That is, a user who was stressed during the morning is more likely to be stressed in the afternoon than a user who was not stressed during the morning. Also, stress at different time points are unlikely to be

identically distributed. For example, the probability that a smoker is stressed on the day before she quits smoking is probably quite different from the probability that the same smoker is stressed on the day after she has quit smoking. However, it may be that the noise level in the dynamics of the context will be sufficiently high so that a model assuming iid contexts and rewards provides a good approximation. Indeed Lei [31] found that in simulated experiments mimicking mobile health studies, the regret of a bandit algorithm similar to those above is robust to dependence between contexts at different times.

2.2 Adversarial Contexts with Stochastic Rewards

The assumption that the contexts are drawn iid from a fixed distribution is quite unrealistic in a lot of practical settings, including mobile health. Researchers have therefore considered a model where the contexts are arbitrary but the reward given context and action is still stochastic in the following sense. Let $\{\mathscr{D}^a(\cdot|x) : x \in \mathscr{X}\}$, for $a \in \{0,1\}$, be two families of distributions over rewards indexed by the context x. Note that we are considering the case of two actions, i.e., $\mathscr{A} = \{0,1\}$. The following online protocol is followed. The contexts are denoted by lower case letters to emphasize that they are not random variables but from an arbitrary deterministic sequence.

1: nature generates $\{x_t\}_{t=1}^T$ in advance
2: **for** $t = 1$ to T **do**
3: receive context x_t
4: algorithm takes action A_t
5: receive reward R_t which is drawn from $\mathscr{D}^{A_t}(\cdot|x_t)$
6: **end for**

Let $\eta_a(x)$ be the expected value of the distribution $\mathscr{D}_a(\cdot|x)$. The optimal policy is given by:

$$\pi^\star(x) = \operatorname*{argmax}_{a \in \mathscr{A}} \eta_a(x),$$

and we define the expected regret of an online learning algorithm as:

$$\sum_{t=1}^T \eta_{\pi^\star(x_t)}(x_t) - \sum_{t=1}^T \mathbb{E}[R_t].$$

All regret bounds mentioned in this subsection hold uniformly over all possible sequences $\{x_t\}_{t=1}^T$ of contexts (with some mild restrictions like boundedness of the contexts).

Li et al. [17] gave an algorithm called LinUCB that is based on the following linearity assumption:

$$\eta_a(x) = \beta_a^\top x, \tag{7}$$

where $x, \beta_a \in \mathbb{R}^p$. LinUCB is here presented as Algorithm 4. It follows a long line of work in bandit algorithms that use upper confidence bounds for action selection. To each action's current estimate, it adds a confidence term which reflects the algorithm's current uncertainty about that estimate. The action selected is the one that maximizes the sum of the estimated reward and the confidence bound.

Algorithm 4 LinUCB Algorithm [2]

Inputs: α (tuning parameter used in computing upper confidence bounds)
$\mathbf{A}^a = \mathbf{I}_{p\times p}, \mathbf{b}^a = \mathbf{0}_{p\times 1}$ for all a

for $t = 1$ to T **do**
  Compute $\hat{\beta}^a = (\mathbf{A}^a)^{-1}\mathbf{b}^a$ for all a // ridge regression
  Compute $U^a = (\hat{\beta}^a)^\top x_t + \alpha\sqrt{x_t^\top (\mathbf{A}^a)^{-1} x_t}$ for all a // upper confidence bound
  Take action $A_t = \text{argmax}_a U^a$ and observe reward R_t
  For $a = A_t$, update $\mathbf{A}^a = \mathbf{A}^a + x_t x_t^\top$, $\mathbf{b}^a = \mathbf{b}^a + R_t x_t$
end for

LinUCB performs well empirically as demonstrated by Li et al. in the context of personalized news article recommendations on the web. However, its theoretical analysis is complicated by the fact that its estimates are not based on iid samples (recall the reward depends on the action and the action is selected using data on prior rewards and prior actions) and there are no known regret bounds. Chu et al. [32] provide an algorithm called SupLinUCB that calls BaseLinUCB as a subroutine and show that it enjoys a regret bound of $O\left(\sqrt{Tp\log^3(T\log T/\delta)}\right)$ with probability at least $1-\delta$. The idea of taking a basic procedure like BaseLinUCB, whose statistical analysis is simplified by assuming independence among the samples, and then using a master algorithm SupLinUCB to ensure the assumption holds, goes back to the work of Auer [33]. His work also considered arbitrary context vectors with linear expected reward functions as in 7 and followed some early line of work in the computer science literature [8, 9, 34, 7]. His basic and master algorithms were called LinRel and SupLinRel. SupLinRel was also shown to enjoy a regret bound of $O(\sqrt{Tp\log^3(T\log T/\delta)})$ with probability at least $1-\delta$. However, LinUCB has practical advantages over LinRel. It is easier to implement and numerically more stable as it relies on ridge regression as its computational core and not on full eigen-decompositions like LinRel. We would like to also point out that LinUCB has been generalized from the standard linear setting as in (7) to the generalized linear setting [35] for use with non-continuous rewards such as binary rewards.

2.2.1 Nonlinear Expected Reward Functions

Readers familiar with the literature on kernel methods and support vector machines in machine learning will recall that these methods deal with non-linearity by em-

bedding the contexts x_t into a high, or even infinite, dimensional space via a feature mapping $\phi(x_t) \in \mathscr{H}_K$, where $\mathscr{H}_K$ is a reproducing kernel Hilbert space (RKHS) corresponding to the kernel $K(x,x') = \langle \phi(x), \phi(x') \rangle$. The kernel K thus measures similarity between contexts using the inner product in a higher dimensional space. LinUCB has been extended to the RKHS setting by Valko et al. [36]. They also provided regret bounds that depend on the "effective dimension" which is, roughly speaking, the number of principal dimensions in which the embedded data points in the RKHS are mostly contained.

Other work on contextual bandits with arbitrary contexts and non-linear expected reward functions includes the Query-ad clustering algorithm of Lu et al. [37] and the RELEAF algorithm of Tekin and van der Schaar [38].

2.2.2 Thompson Sampling

Thompson sampling, also called "posterior sampling" [39] or "probability matching" [40], is a Bayesian approach to designing online learning algorithms for bandit problems. In the linear setup as in (7) above, it involves choosing prior distributions for the unknown reward parameters β_a and choosing conditional distributions for the rewards given context and action. Algorithm 5 chooses the prior to be a multivariate normal distribution with mean zero and covariance matrix $\sigma^2 I_{p\times p}$. It also assumes that the reward given context x and action a is drawn from a normal distribution with mean $\beta_a^\top x$ and variance σ^2. At every time step, it draws samples $\widetilde{\beta}_a$ from the posterior distribution for β_a and chooses the action with the highest mean $\widetilde{\beta}_a^\top x_t$ according to the drawn posterior samples. Once the action is taken and the corresponding reward observed, it updates the posterior distribution for the corresponding reward parameter.

Algorithm 5 Thompson Sampling Algorithm [2]

Inputs: σ^2 (variance parameter used in the prior and in the reward linear model)
$\mathbf{A}^a = \mathbf{I}_{p\times p}, \mathbf{b}^a = \mathbf{0}_{p\times 1}$ for all a

for $t = 1$ to T **do**
  Compute $\hat{\beta}^a = (\mathbf{A}^a)^{-1}\mathbf{b}^a$ for all a
  Sample $\widetilde{\beta}^a$ from $\text{NORMAL}(\hat{\beta}^a, \sigma^2(\mathbf{A}^a)^{-1})$ for all a // Sample from the posterior
  Take action $A_t = \text{argmax}_a(\widetilde{\beta}^a)^\top x_t$ and observe reward R_t
  For $a = A_t$, update $\mathbf{A}^a = \mathbf{A}^a + x_t x_t^\top$, $\mathbf{b}^a = \mathbf{b}^a + R_t x_t$
end for

Agrawal and Goyal [41] analyze Algorithm 5 and prove a regret bound of $O\left(p\sqrt{\frac{T^{1+\varepsilon}}{\varepsilon}(\log T \log(1/\delta))}\right)$ with probability $1-\delta$. Here $\varepsilon \in (0,1)$ is a tuning parameter. Thompson sampling had been applied to contextual bandits [42] before Agrawal and Goyal's work but finite time regret bounds were not available. Agrawal

and Goyal's regret analysis holds under much weaker assumptions that made to derive the Thompson Sampling algorithm itself. First, the regret analysis itself makes no use of the prior. It holds for every β_a choice as long as it is bounded. Second, it does not assume that the rewards are actually drawn from a normal distribution. It does require the linearity assumption in (7) to hold but the rewards are only assumed to be sub-gaussian.

As discussed above, this section does not require that the contexts are iid. Thus the bandit algorithms considered here can accommodate settings in which the contexts can have arbitrary relationships one with another. Despite this, as discussed above, for some algorithms one can guarantee how fast the algorithm learns with time. This may be useful in mobile health particularly in areas of science where the dynamic evolution of the contexts over time are not yet well understood, for example when the context includes craving for substances or alternately physiological and perceived stress. However, this setting continues to be potentially problematic in that how users respond to interventions (e.g., reward distribution given context) can change with time. For example, the relationship between self-efficacy and relapse to smoking appears to change as time increases from the quit date [43]; this is likely to mean that the distribution of the reward as a function of an intervention option and a context involving self-efficacy is likely to change with time as well.

2.3 Fully Adversarial Contextual Bandits

In this section, we further relax our assumptions on how the contexts and rewards are generated. First, we consider a setting where the adversary chooses a sequence of contexts and reward *distributions*. In this setting, the aim is do well with respect to a policy that knows the sequence of distributions in advance. Second, we consider a setting where the adversary chooses a sequence of contexts and reward *values*. In this setting, the aim is do well with respect to a pre-defined class Π of policies.

2.3.1 Competing against Greedy Policies with Changing Reward Distributions

Here the context sequence as well as the sequence of reward distributions given context and action are chosen arbitrarily. Denote the choice of the action a's reward distribution given context x at time t by $\mathscr{D}_t^a(\cdot|x)$. Denote the expected reward under this distribution by $\eta_t^a(x)$. Consider the following online protocol.

1: nature generates $\{(x_t, \mathscr{D}_t^0(\cdot|x), \mathscr{D}_t^1(\cdot|x))\}_{t=1}^T$ in advance
2: **for** $t = 1$ to T **do**
3: receive context x_t
4: algorithm takes action A_t
5: receive reward R_t drawn from the distribution $\mathscr{D}^{A_t}(\cdot|x_t)$ with expectation $\eta_t^{A_t}(x_t)$
6: **end for**

At the end of T rounds, the time-average of the expected reward functions for action a played by nature is $\bar{\eta}^a(x) = \frac{1}{T}\sum_{t=1}^{T} \eta_t^a(x)$. The regret definition below compares the learning algorithm's expected reward to that of the greedy policy with respect to $\bar{\eta}^a$:

$$\pi^\star(x) = \textsc{Greedy}(\bar{\eta}^a)(x) = \operatorname*{argmax}_{a\in\mathscr{A}} \sum_{t=1}^{T} \eta_t^a(x).$$

Regret is now defined as

$$\sum_{t=1}^{T} \eta_t^{\pi^\star(x_t)}(x_t) - \sum_{t=1}^{T} \mathbb{E}[R_t].$$

Note that in the protocol above, there are two sources of randomness. First, there is randomness in nature's realization of the rewards unless the distributions $\mathscr{D}_t^a(\cdot|x)$ are point masses. Second, the online learning algorithm may be a randomized one and could be using additional randomization to select its actions A_t. The expectation above is with respect both possible sources of randomness.

We know of only one paper that gives bandit algorithms in this setting. Slivkins [20] considers this setting in Section 7 of his paper. In this chapter, we have mostly focused on the case of two actions, i.e., $\mathscr{A} = \{0,1\}$ and our context space $\mathscr{X}$ has often been a subset of $\mathbb{R}^p$. He considers a much more general case when $\mathscr{A}, \mathscr{X}$ are metric spaces. In our specific setting, his assumptions on the expected reward functions is that they are Lipschitz with respect to some norm $\|\cdot\|$ defined on $\mathbb{R}^p$:

$$\forall t, x, x', a, a',\ |\eta_t^a(x) - \eta_t^a(x')| \leq \|x - x'\|.$$

His algorithm achieves a regret bound of $O(T^{1-1/(2+d_{\mathscr{X}})}(\log T))$ where $d_{\mathscr{X}}$ is the covering dimension of the context space $\mathscr{X}$ under the metric $\|x-x'\|$. Note that the covering dimension of a metric space is defined as the smallest integer d such that the number of balls of radius r required to cover the space is $O(r^{-d})$. When $\mathscr{X} \subseteq \mathbb{R}^p$ is a bounded set, we always have $d_{\mathscr{X}} \leq p$.

2.3.2 Competing against a Fixed Class of Policies

In the previous section, the policy we compete against was indirectly defined by the expected reward functions played by nature. Here we fix a class Π of policies in advance and try to compete with the best policy in Π. The protocol is now as follows. Note that now nature generates arbitrary contexts and reward values for the two actions.

1: nature generates $\{(x_t, r_t^0, r_t^1)\}_{t=1}^T$ in advance
2: **for** $t = 1$ to T **do**
3: receive context x_t
4: algorithm takes action A_t

5: receive reward $R_t = r_t^{A_t}$
6: **end for**

Regret is now defined as

$$\max_{\pi \in \Pi} \sum_{t=1}^{T} r_t^{\pi(x_t)} - \sum_{t=1}^{T} \mathbb{E}[R_t].$$

Regret bounds in the adversarial setting hold uniformly over all choices of the context, reward sequence $\{(x_t, r_t^0, r_t^1)\}_{t=1}^T$.

A special case of the above setup when there is only one unchanging context, $x_t = x$, reduces to the adversarial multi-armed bandit problem with K arms (we have focused on the $K = 2$ case in this chapter). This problem was first considered by Auer et al. [44]. Their Exp3 algorithm obtains an expected regret bound of $O(\sqrt{KT \log K})$ which can be improved to $O(\sqrt{KT})$ using a different algorithm [45, 46]. They also present a variant Exp3.P that enjoys a bound on the regret not just in expectation but with high probability. More interesting in the contextual bandit setting is their Exp4 algorithm. Exp4 applies in the case when there are a finite number of "experts" each suggesting an action to take at a given round. We can identify their experts with policies in the set Π if the set is finite. We present the Exp4 algorithm as Algorithm 6. They prove an expected regret bound of $O(\sqrt{KT \log(|\Pi|)})$ for Exp4 which reduces to $O(\sqrt{T \log |\Pi|})$ when $K = 2$. Even though the regret bound can tolerate very large policy classes, the implementation of the algorithm itself is practical only for very small policy classes since Exp4 maintains a weight for each policy in the class.

Algorithm 6 Exp4 Algorithm [2]

Inputs: $\gamma \in (0, 1]$ (learning rate/step size; also used an an exploration parameter)

$w_\pi = 1$ for all $\pi \in \Pi$ // set equal weights for all policies initially
for $t = 1$ to T **do**
  Compute $W = \sum_{\pi \in \Pi} w_\pi$

  /$\star$ convert policy weight into action probabilities $\star$/
  For all a, compute $p_a = (1 - \gamma)\frac{1}{W} \sum_{\pi \in \Pi} w_\pi \mathbf{1}[\pi(x_t) = a] + \gamma/2$

  Choose $A_t = a$ with probability p_a and observe reward R_t
  Set $\hat{r}^a = R_t / p_a$ if $A_t = a$ and 0 otherwise // estimate rewards for both actions
  For all $\pi \in \Pi$, set $w_\pi = w_\pi \exp\left(\gamma \hat{r}^{\pi(x_t)}/2\right)$ // update policy weights
end for

High probability guarantees matching those of Exp4 have been obtained by Beygelzimer et al. [47] using their Exp4.P algorithm. Note that the same paper also presents an algorithm VE for the stochastic setting when the context and reward tuples are drawn iid from a fixed distribution. Even if Π is an infinite class but the VC

dimension of Π is $d < \infty$, VE enjoys a regret bound of $O(\sqrt{dT\log(T/(d\delta))})$ with probability at least $1-\delta$.

At least conceptually, the Exp family algorithms provided in this section appear rather promising because they require the least restrictive assumptions on the rewards in order to learn. However these algorithms, because they are designed to work in the worst cases, may learn too slowly for a large subset of a particular population such as the population of smokers who are trying to quit. At this time, we do not have good rules of thumb for selecting the type of algorithm to employ for optimizing mobile health interventions depending on the type of populations and behavior change problem.

3 Challenges in Mobile Health Applications

We have seen that a wide variety of contextual bandit algorithms and theoretical frameworks to analyze them already exist in the literature. These ideas serve as useful starting points for the design of online learning algorithms in mobile health. However, to truly make an impact in mobile health, significant work needs to be done to deal with challenges that arise in the mobile health setting. In this section, we consider some of these challenges and explore ways to start addressing them.

3.1 Finding a Good Initial Policy

Good initialization of the learning process is very important. If the algorithm chooses very bad actions in the beginning, it can have a negative impact on health outcomes and user engagement. One possibility is to consult domain experts and use an expert derived policy at the start. However, it might turn out to be difficult to turn intuitive judgements of domain experts into a precisely stated policy. Moreover, mobile health is a relatively new area and often domain experts lack sufficient knowledge of what works and what does not when interventions are delivered through mobile devices and wearables.

We think that it is much better to proceed in an evidence-based manner and initialize the policy using data previously gathered, say in a microrandomized trial [48]. Data from a microrandomized trial can be used for a variety of purposes including estimation of the value of a policy in question. If candidate policies can be evaluated then a good one can be selected from a set of policies. Microrandomized trials offer very high-quality data. But even less high quality data can be useful. For example, if the policy that generated data in a mobile health study is exactly or partially known then one can still form reasonable estimates of the value of a given policy. The problem of using an existing batch of data gathered under one policy to reason about the value of another policy is called the problem of "offline learning" or "offline evalu-

ation". There is work in both the computer science [49, 50, 51, 52, 53], as well as the statistics literature on this problem [54, 55, 56, 57].

3.2 Interpretability of the Learned Policy

Progress in mobile health will occur when human-computer interface researchers, machine learning researchers and statisticians work in close collaboration with domain scientists such as behavioral scientists. On the one hand, we need guidance from theories of behavior change to guide the development of mobile health interventions. On the other hand, the policy learned using online learning algorithms needs to be communicated back to behavioral scientists so that they can interpret it in light of their theories or use it to change and refine existing theories. This communication is facilitated by learning interpretable policies. Using policies represented by large decision trees, deep neural networks or kernel methods may not lend themselves easily to interpretation.

Lei [31] has explored the use of actor-critic methods from the reinforcement learning literature in setting of contextual bandits. The critic part is responsible for estimating the expected reward function and can use very flexible non-parametric and non-linear regression methods. The actor part is responsible for generating a policy using the estimates provided by the critic. Since only the policy needs to be communicated to the domain scientist, we just need to keep the actor architecture simple by choosing a low-dimensional interpretable policy parameterization.

3.3 Assessing Usefulness of Contextual Variables

Contextual variables in mobile health are often costly to acquire. If they are passively sensed by the phone (e.g., GPS location) or a wearable (e.g., heartrate), acquiring them drains the battery. If they are actively acquired by asking the user a self-report question (e.g., about their mood), acquiring them incurs user burden. Therefore, it is important to develop methods that enable researchers to decide whether or not a contextual variable is useful for deciding which intervention to deliver. For example, suppose we use the following interpretable parameterization for a stochastic policy

$$\pi(x) = \frac{\exp(\beta^\top x)}{1+\exp(\beta^\top x)}.$$

Note that π maps the context to $[0,1]$ instead of $\{0,1\}$ and should be interpreted as the *probability* to taking action 1. This is called the "Gibbs" or the "expit" parameterization. If we simply output an estimate $\widehat{\beta}$ at the end of the learning process, it is not very useful for assessing usefulness of variables. We need to provide confidence intervals for these estimates. Then, we can see whether a 95% confidence

interval for, say, $\widehat{\beta}_1$, contains zero or not. This will provide researchers with an evidence-based method to decide whether the first contextual variable in the context x is useful or not. We have not seen many tools to enable such reasoning in existing contextual bandit algorithms. An exception is the work of Lei [31] mentioned above that does construct confidence intervals for the policy parameters estimated using their actor-critic online learning algorithm.

3.4 Computational Considerations

Computation on mobile phones consumes resources. If we perform computations on the phone we need to think about implementing the learning algorithms very efficiently in order to not put an undue burden on the phone's performance and battery life. If we perform computations on the cloud, we need to minimize data transfer between the phone and the cloud to save the phone's resources. We also need to take into account occasional failures, due to a bad network reception or drained battery. These failures can cause the learning algorithm to not be able to push fresh data to the cloud or pull the latest policy or action recommendation from the cloud. There is little work on designing and proving guarantees about contextual bandit algorithms that are resilient to such failures.

Another question that needs work is how to tradeoff the frequency of learning with the noise level in the data. All algorithms presented above make an update whenever an action is selected and a reward is observed. If the data is very noisy then we might have the learning algorithm update its policy at larger time intervals so as to acquire more information. What should be the time intervals at which our learning algorithm updates the policy? To answer this question, one will have to consider the computational complexity of the update as well the amount (governed perhaps by a step size parameter) by which a single update changes the policy.

3.5 Robustness to Failure of Assumptions

Algorithms designed for the worst-case adversarial framework can perform suboptimally when data is actually generated stochastically. Algorithms that have guarantees under stochastic assumptions can behave badly when the specific stochastic assumptions underlying their analysis are not met. In mobile health, where the consequence of such non-robustness is worse health outcomes for people, we need to pay serious attention to such issues. Three assumptions that make repeated appearance in the theoretical analyses of contextual bandit algorithms are independence, stationarity, and absence of the impact of actions on the user's future contexts. Any candidate online learning algorithm needs to be tested for reasonable departures from these ideal assumptions in simulations before being deployed in a real study with users. Existing algorithms need to be analyzed under weaker assumptions, if

possible. Otherwise, attempts should be made to quantify the degradation in performance in non-ideal settings. New algorithms that are more robust to failure of assumptions need to be designed and associated guarantees provided.

Some contextual bandit algorithms enjoy regret guarantees only in expectation. But an algorithm whose regret is small in expectation, but has high variance, can have very serious consequences in mobile health. High variance in regret means that occasionally, the algorithm performs very poorly and its regret is much larger than the provided guarantee. The will translate into adverse health outcomes for some people in the cohort being studied. High probability guarantees on the regret are better than guarantees in expectation but they are simply the first step in the direction of designing learning algorithms that better manage the risk of hurting people's health outcomes. There is some work on risk-aversion in multi-armed bandit problems [58, 59]. It is possible that some of the techniques developed there can be useful for contextual bandit learning algorithms too.

3.6 Costly to Acquire or Missing Contexts and Rewards

As noted above, contextual variables can be costly to acquire in a mobile health setting. Even rewards can be costly to acquire especially if they cannot be passively sensed and we have to rely on user self-reports. If a variable is indeed useful for decision-making then choosing not to acquire it will lead to sub-optimal decisions. Similarly, we cannot simply decide to not acquire the reward variable because doing so will hamper the ability of the learning algorithm to learn from observed rewards. The key is to acquire costly variables judiciously. We can maintain predictions of such variables and acquire them only when uncertainty about them increases beyond a threshold. If the costs associated with acquisition can be quantified then it can be formally included in the definition of regret. Currently, we do not have much guidance on how to deal with costly to acquire contexts and rewards.

Another aspect not treated properly in the existing literature is missingness of contextual variables and rewards. Maintaining predictions of variables that can be potentially missing, of course, helps. However, missingness of self-reported data can also indicate one or more of the following: high user stress, high user busyness and low user engagement. Thus, missingness can itself be used as a contextual variable to use in decision-making. More research is needed to fully integrate support for missing data in existing contextual bandit algorithms.

References

1. Michael Woodroofe. A one-armed bandit problem with a concomitant variable. *Journal of the American Statistical Association*, 74(368):799–806, 1979.
2. John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824,

2008.
3. John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.
4. Chih-Chun Wang, Sanjeev R. Kulkarni, and H. Vincent Poor. Bandit problems with side observations. *Automatic Control, IEEE Transactions on*, 50(3):338–355, 2005.
5. Chih-Chun Wang, Sanjeev R. Kulkarni, and H. Vincent Poor. Arbitrary side observations in bandit problems. *Advances in Applied Mathematics*, 34(4):903–938, 2005.
6. Alexander Goldenshluger and Assaf Zeevi. A note on performance limitations in bandit problems with side information. *Information Theory, IEEE Transactions on*, 57(3):1707–1713, 2011.
7. Naoki Abe and Philip M. Long. Associative reinforcement learning using linear probabilistic concepts. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 3–11, 1999.
8. Leslie P. Kaelbling. Associative reinforcement learning: A generate and test algorithm. *Machine Learning*, 15(3):299–319, 1994.
9. Leslie P. Kaelbling. Associative reinforcement learning: Functions in k-DNF. *Machine Learning*, 15(3):279–298, 1994.
10. Naoki Abe, Alan W. Biermann, and Philip M. Long. Reinforcement learning with immediate rewards and linear hypotheses. *Algorithmica*, 37(4):263–293, 2003.
11. Alexander L. Strehl, Chris Mesterharm, Michael L. Littman, and Haym Hirsh. Experience-efficient learning in associative bandit problems. In *Proceedings of the 23rd international conference on Machine learning*, pages 889–896. ACM, 2006.
12. Murray K. Clayton. Covariate models for Bernoulli bandits. *Sequential Analysis*, 8(4):405–426, 1989.
13. Jyotirmoy Sarkar. One-armed bandit problems with covariates. *The Annals of Statistics*, pages 1978–2002, 1991.
14. Yuhong Yang and Dan Zhu. Randomized allocation with nonparametric estimation for a multi-armed bandit problem with covariates. *The Annals of Statistics*, 30(1):100–121, 2002.
15. Philippe Rigollet and Assaf Zeevi. Nonparametric bandits with covariates. In Adam Tauman Kalai and Mehryar Mohri, editors, *Proceedings of the 23rd Conference on Learning Theory*, pages 54–66, 2010.
16. Naoki Abe and Atsuyoshi Nakamura. Learning to optimally schedule internet banner advertisements. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 12–21. Morgan Kaufmann Publishers Inc., 1999.
17. Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670. ACM, 2010.
18. Inbal Nahum-Shani, Shawna N. Smith, Bonnie J. Spring, Linda M. Collins, Katie Witkiewitz, Ambuj Tewari, and Susan A. Murphy. Just-in-time adaptive interventions (JITAIs) in mobile health: Key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine*, 2016. accepted subject to revisions.
19. Yevgeny Seldin, Peter Auer, John S. Shawe-Taylor, Ronald Ortner, and François Laviolette. PAC-Bayesian analysis of contextual bandits. In *Advances in Neural Information Processing Systems*, pages 1683–1691, 2011.
20. Aleksandrs Slivkins. Contextual bandits with similarity information. *The Journal of Machine Learning Research*, 15(1):2533–2568, 2014.
21. Rajeev Agrawal and Demosthenis Teneketzis. Certainty equivalence control with forcing: revisited. *Systems & control letters*, 13(5):405–412, 1989.
22. Alexander Goldenshluger and Assaf Zeevi. A linear response bandit problem. *Stochastic Systems*, 3(1):230–261, 2013.
23. Alexander Goldenshluger and Assaf Zeevi. Woodroofe's one-armed bandit problem revisited. *The Annals of Applied Probability*, 19(4):1603–1633, 2009.
24. Hamsa Bastani and Mohsen Bayati. Online decision-making with high-dimensional covariates. *Available at SSRN 2661896*, 2015.

25. Alekh Agarwal, Miroslav Dudík, Satyen Kale, John Langford, and Robert E. Schapire. Contextual bandit learning with predictable rewards. In *International Conference on Artificial Intelligence and Statistics*, pages 19–26, 2012.
26. Vianney Perchet and Philippe Rigollet. The multi-armed bandit problem with covariates. *The Annals of Statistics*, 41(2):693–721, 2013.
27. Wei Qian and Yuhong Yang. Randomized allocation with arm elimination in a bandit problem with covariates. *Electronic Journal of Statistics*, 10(1):242–270, 2016.
28. Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence*, pages 169–178. AUAI Press, 2011.
29. Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1638–1646, 2014.
30. Consumer Health Information Corporation. Motivating patients to use smartphone health apps, 2011. URL: http://www.consumer-health.com/motivating-patients-to-use-smartphone-health-apps/, accessed: June 30, 2016.
31. Huitian Lei. *An Online Actor Critic Algorithm and a Statistical Decision Procedure for Personalizing Intervention*. PhD thesis, University of Michigan, 2016.
32. Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.
33. Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research*, 3:397–422, 2003.
34. Philip M. Long. On-line evaluation and prediction using linear functions. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 21–31. ACM, 1997.
35. Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594, 2010.
36. Michal Valko, Nathan Korda, Rémi Munos, Ilias Flaounas, and Nello Cristianini. Finite-time analysis of kernelised contextual bandits. In *Uncertainty in Artificial Intelligence*, page 654, 2013.
37. Tyler Lu, Dávid Pál, and Martin Pál. Contextual multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 485–492, 2010.
38. Cem Tekin and Mihaela van der Schaar. RELEAF: An algorithm for learning and exploiting relevance. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):716–727, June 2015.
39. Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.
40. Steven L. Scott. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658, 2010.
41. Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 127–135, 2013.
42. Benedict C. May, Nathan Korda, Anthony Lee, and David S. Leslie. Optimistic Bayesian sampling in contextual-bandit problems. *The Journal of Machine Learning Research*, 13(1):2069–2106, 2012.
43. Saul Shiffman. Dynamic influences on smoking relapse process. *Journal of Personality*, 73(6):1715–1748, 2005.
44. Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.
45. Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2004.
46. Jacob Abernethy, Chansoo Lee, and Ambuj Tewari. Fighting bandits with a new kind of smoothness. In *Advances in Neural Information Processing Systems 28*, pages 2188–2196, 2015.

47. Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, volume 15 of *JMLR Workshop and Conference Proceedings*, pages 19–26, 2011.
48. Predrag Klasnja, Eric B. Hekler, Saul Shiffman, Audrey Boruvka, Daniel Almirall, Ambuj Tewari, and Susan A. Murphy. Microrandomized trials: An experimental design for developing just-in-time adaptive interventions. *Health Psychology*, 34(Suppl):1220–1228, Dec 2015.
49. John Langford, Alexander Strehl, and Jennifer Wortman. Exploration scavenging. In *Proceedings of the 25th international conference on Machine learning*, pages 528–535. ACM, 2008.
50. Alex Strehl, John Langford, Lihong Li, and Sham M. Kakade. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Systems*, pages 2217–2225, 2010.
51. Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306. ACM, 2011.
52. Lihong Li, Wei Chu, John Langford, Taesup Moon, and Xuanhui Wang. An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. In *Proceedings of the Workshop on On-line Trading of Exploration and Exploitation 2 July 2, 2011, Bellevue, Washington, USA*, volume 26 of *JMLR Workshop and Conference Proceedings*, pages 19–36, 2012.
53. Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1097–1104, 2011.
54. Min Qian and Susan A. Murphy. Performance guarantees for individualized treatment rules. *Annals of Statistics*, 39(2):1180, 2011.
55. Yingqi Zhao, Donglin Zeng, A. John Rush, and Michael R. Kosorok. Estimating individualized treatment rules using outcome weighted learning. *Journal of the American Statistical Association*, 107(499):1106–1118, 2012.
56. Baqun Zhang, Anastasios A. Tsiatis, Eric B. Laber, and Marie Davidian. A robust method for estimating optimal treatment regimes. *Biometrics*, 68(4):1010–1018, 2012.
57. Baqun Zhang, Anastasios A. Tsiatis, Marie Davidian, Min Zhang, and Eric Laber. Estimating optimal treatment regimes from a classification perspective. *Stat*, 1(1):103–114, 2012.
58. Amir Sani, Alessandro Lazaric, and Rémi Munos. Risk-aversion in multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 3275–3283, 2012.
59. Sattar Vakili and Qing Zhao. Mean-variance and value at risk in multi-armed bandit problems. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1330–1335. IEEE, 2015.