Combining Offline Causal Inference and Online Bandit Learning for Data Driven Decisions

Li Ye
The Chinese University of Hong Kong

Yishi Lin
Tencent

Hong Xie
College of Computer Science, Chongqing University

John C.S. Lui
The Chinese University of Hong Kong

ABSTRACT

A fundamental question for companies is: *How to make good decisions with the increasing amount of logged data?*. Currently, companies are doing online tests (e.g. A/B tests) before making decisions. However, online tests can be expensive because testing inferior decisions hurt users' experiences. On the other hand, offline causal inference analyzes logged data alone to make decisions, but once a wrong decision is made by the offline causal inference, this wrong decision will continuously to hurt all users' experience. In this paper, we unify offline causal inference and online bandit learning to make the right decision. Our framework is flexible to incorporate various causal inference methods (e.g. matching, weighting) and online bandit methods (e.g. UCB, LinUCB). For these novel combination of algorithms, we derive theoretical bounds on the decision maker's *"regret"* compared to its optimal decision. We also derive the first regret bound for forest-based online bandit algorithms. Experiments on synthetic data show that our algorithms outperform methods that use only the logged data or only the online feedbacks.

1 INTRODUCTION

How to make the right decision is the key challenge in many web applications. For example, recommender systems need to decide which item to recommend for each user. Sellers in eBay-like E-commerce systems need to determine the price for their products. An Internet company that sells in-feeds advertisements (ads.) needs to decide whether to place an ad. below a video (or below other contents such as images and texts), as illustrated in Figure 1.

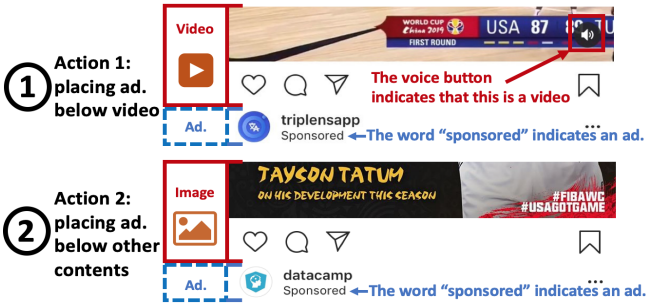

Figure 1: In-feeds advertisement placement of Instagram

It is common that companies for the above web applications have some logged data which may assist the decision making, e.g., a recommender system has logs of its recommendations and users' reactions. An Internet company which sells in-feeds advertisements has logs of advertisement placement and users' click feedbacks as shown in Table 1. The question is: *can we use these logged data to make a better decision?* To illustrate, consider the following example.

Table 1: Example of logged data of a company selling ads.

	action	*contexts*			*outcome*
User id	**Video above the ad?**	**Does the user like videos?**	**Age**	**···**	**Click?**
1	no	no	30	···	no (0)
2	yes	yes	20	···	yes (1)

Example 1. *There are 10,000 incoming users who will arrive to see the advertisement. The Internet company needs to decide whether to place the ad. below a video (or below other contents). The objective is to attract more clicks from these 10,000 new users, as one click yields a revenue of $1. The users are categorized into two types: who "like" or "dislike" videos. For simplicity, suppose with a probability 0.5, each user likes (or dislike) videos. The true click rates for each type of user, which are unknown to the company, are summarized in Table 2. Suppose we have a logged data of 400 users, and half of them like (or dislike) watching videos. The click rates are summarized in Table 3.*

Table 2: Setting - true click rates of each types of users

Expected click rate	**Like videos**	**Dislike videos**
Video above ad.	*11%*	*1%*
No video above ad.	*14%*	*4%*

Table 3: A summary statistics of logged data of 400 users

Average click rate	**Like videos**	**Dislike videos**
Video above ad.	*10% from 150 ads.*	*2% from 50 ads.*
No video above ad.	*12% from 50 ads.*	*4% from 150 ads.*

In the logged data (as shown in Table 3), the users who like videos are more likely to see videos above ad., because they subscribe to more video producers. Let us start with the following simple decision strategy for the decision problem described in Example 1.

Strategy 1 (Empirical average of the logged data). *The company uses the empirical average click rates for each action from the logged data. Then the company selects the action with the higher click rate for all the incoming 10,000 users in Example 1.*

Applying Strategy 1 to the logged data in Table 3, one can compute the average click rate when the ad. was placed below a video as

$$(10\% \times 150 + 2\% \times 50)/(150 + 50) = 8\%.$$

Similarly, the click rate for the action of not placing the ad. below a video is 6%. Thus, the Internet company selects the action of placing the ad. below a video. However, as shown in Table 2, the optimal action is *not* to place the ad. below a video for both types of users.

Strategy 1 fails because it ignores users' *preferences to videos* in the comparison. One alternative method to mitigate this issue is to use the following offline causal inference strategy[29][34].

Strategy 2 (Offline causal inference). *First, compute the average click rates with respect to each user type (done in Table 3). Second, for each action, compute the weighted average of these click rates, where the weight is the number of the users of each type. Finally, select the action with the higher click rates for all the 10,000 incoming users.*

Applying strategy 2 to the logged data in Table 3, the average click rate for the action of placing the ad. below a video is

$$10\% \times (200/400) + 2\% \times (200/400) = 6\%.$$

Similarly, users' average click rate for the action of not placing the ad. below a video is $(12\% + 4\%)/2 = 8\%$. Strategy 2 selects the action of not placing the ad. below a video, which is optimal in Example 1. However, the causal inference strategy has a risk of not finding the optimal action because the logged data is finite (since the logged data were samples from the population). For example, it is possible that the company collects the logged data in Table 4. For this logged data, the causal inference strategy will select the inferior action of placing the ad. below a video, for all 10,000 incoming users.

Table 4: Another possible logged data of 400 users

Average click rate	Like videos	Dislike videos
Video above ad.	10% from 150 ads.	4% from 50 ads.
No video above ad.	8% from 50 ads.	4% from 150 ads.

Strategy 2 has a risk of making a wrong decision because the sample size of logged data is finite and it does not adjust according to online feedbacks. On the other hand, Strategy 1 fails because it uses incomplete data which do not have the important aspect of users' *preferences to videos*. One popular way to mitigate these issues is to use the online A/B test strategy[39].

Strategy 3 (Online A/B test). *Each of the first 4,000 incoming users is randomly distributed to group A or B with an equal probability. Users in group A see a video above the advertisement, while users in group B do not. Then, the company chooses the action in the group that has a higher average click rate for the remaining 6,000 users.*

The above online A/B test method gradually finds the optimal action as the number of testing users becomes large. However, this is achieved at *a high cost* of testing the inferior action. Around 2,000 (out of 10,000) users will be distributed to the group A that tests the inferior action with a click rate 6% (instead of the optimal 9%).

Motivated by the above pros and cons of three typical decision making strategies, we aim to answer the following questions:
(1) *How to make decisions when the decision maker has logged data?*
(2) *Can we design algorithms that make "nearly-optimal" decisions?*
To answer these questions, we propose the following strategy:

Strategy 4 (Causal inference + online learning (ours)). *Use the offline causal inference strategy to construct confidence bounds of click rates for different actions. Then, use the online learning, say Upper Confidence Bound algorithm[6], to make online decisions.*

To illustrate the benefits of strategy 4, consider Example 1 with the logged data generated according to the same probability distributions as shown in Table 2. Table 5 presents the company's expected revenue under the above four strategies. We call the difference between the optimal revenue ($900 in this example) and a strategy's revenue as the *"regret"* of the strategy. Table 5 shows that our strategy has the highest revenue (or the lowest regret).

Table 5: The expected revenue ($) of the four strategies over 10,000 users. The optimal is $900 (with no vidoes above ads)

Strategy	Empirical average	Causal inference	Online A/B Test	Ours
Expected revenue	674.4	847.7	868.8	**894.4**
Regret to optimal	225.6	52.3	60.1	**5.6**

In practice, the logged data or the online decision making setting can be much more complicated than that in Example 1. For example, there may be some unobserved contexts so the logged data can not represent the online environment. Such mis-matched logged data can mislead the decision maker. Furthermore, even when the logged data are representative samples of the online environment, it can still mislead the decision maker due to its limited sample size and the large variation of users' behaviors. These factors make it challenging to design a good strategy. Our contributions are:

- **A unified framework and novel algorithms.** We formulate a general online decision making problem with logged data, where we consider both population-level and individualized decisions. Then, we provide an algorithmic framework to unify both offline causal inference and online bandit learning. It uses both the logged data and online feedbacks. We then get novel instantiations of algorithms by combining various causal inference methods like matching and weighting[8], as well as various bandit algorithms like UCB[6] and LinUCB[27]. This unification inspires us to extend the random-forest algorithm for causal inference to the *"ϵ-greedy causal-forest"* online decision algorithm.
- **New theorecical regret bounds.** We develop a framework to analyze the regret for algorithms using our framework. We show how the regret bound decreases as the amount of logged data increases. When there is no logged data, our regret bound reduces to that of online bandit algorithms. When the data amount is sufficiently large (w.r.t. time T), our algorithms achieve constant cumulative regrets. In addition, we derive an asymptotic regret bound for the "ϵ-greedy causal-forest" algorithm. To the best of our knowledge, this is the first regret analysis for a forest-based bandit algorithm.
- **Experiments on synthetic datasets.** We evaluate our algorithms on synthetic environment. Experiments on both the synthetic data demonstrate that our algorithms that use both logged data and online feedbacks achieve the highest reward for the company. Our experiments show that when the data do not follow the linear properties, our ϵ-greedy causal-forest algorithm can still achieve twice the rewards compared to LinUCB[27].

2 MODEL & PROBLEM FORMULATION

We first present our logged data model. Then we model the online environment and present its connections to the logged data. Finally we present the online decision problem which aims to utilize both the logged data and online feedbacks to minimize the regret.

2.1 The Logged Data Model

We consider a tabular logged dataset (e.g., Table 1), which was collected before the running of online testing algorithms. The logged dataset has $I \in \mathbb{N}_+$ data items, denoted by

$$\mathcal{L} \triangleq \{(a_i, \boldsymbol{x}_i, y_i) | i \in [-I]\},$$

where $[-I] \triangleq \{-I, -I+1, \ldots, -1\}$ and $(a_i, \boldsymbol{x}_i, y_i)$ denotes the i^{th} recorded data item. Here, we use negative indices to indicate that the logged data was collected in the past. For example, Table 1 shows a logged dataset with $I = 2$ items. In the logged data, $a_i \in [K] \triangleq \{1, \ldots, K\}$ denotes an action of our interest, where $K \in \mathbb{N}_+$. The action in the logged data can be generated according to the users' natural behaviors or by the company's interventions. In Table 1, we have $K = 2$, where $a_i \in \{1(yes), 2(no)\}$ indicates whether to place the ad. below a video. The $y_i \in \mathcal{Y} \subseteq \mathbb{R}$ denotes the outcome (or reward). In Table 1, we have $\mathcal{Y} = \{0, 1\}$, where y_i indicates whether a user clicks the advertisement. Furthermore, $\boldsymbol{x}_i \triangleq (x_{i,1}, \ldots, x_{i,d}) \in \mathcal{X}$ denotes the context (or feature)[27], where $d \in \mathbb{N}_+$ and $\mathcal{X} \subseteq \mathbb{R}^d$. In Table 1, the $\boldsymbol{x}_i$ can represent a user's age, preference to videos, etc. The context is also called *"observed confounders"* in the causal inference literature[8]. We use $\boldsymbol{u}_i \triangleq (u_{i,1}, \ldots, u_{i,\ell}) \in \mathcal{U}$, where $\ell \in \mathbb{N}_+$ and $\mathcal{U} \subseteq \mathbb{R}^\ell$, to model the unobserved confounders. The $\boldsymbol{u}_i$ captures latent or hidden contexts, for example whether a user is prone to the social influence. We consider the general case that the contexts in logged data $\{\boldsymbol{x}_i\}_{i \in [-I]}$ can be non-random observations, i.e., may not be generated from a probability distribution.

Now we introduce the underlying probability law of the data generation process. For the i^{th} user with context $\boldsymbol{x}_i$, let A_i denote the random variable that generates the action a_i. To capture the uncertainty of the outcome, let the random variable $Y_i(k)$ denote the outcome for the i^{th} user if we had changed the action of the i^{th} user to k. When $k \neq a_i$, $Y_i(k)$ is a "potential outcome" in the Rubin's causal model[32] that cannot be observed in the logged data. We have the following two assumptions on potential outcomes, which are standard assumptions in the causal inference literature[32]. The first assumption states that the potential outcome of a data item is independent of the actions of other data items.

Assumption 1 (Stable unit for logged data). *One data item's potential outcome is independent of the actions of other data items:*

$$\mathbb{P}[Y_i(k)=y | A_i=a_i, A_j=a_j] = \mathbb{P}[Y_i(k)=y | A_i=a_i], \forall y \in \mathcal{Y}, j \neq i. \quad (1)$$

The following assumption captures that for each data item, the potential outcomes are independent of the action given the context.

Assumption 2 (Ignorability). *The potential outcomes satisfy*

$$[Y_i(1), \ldots, Y_i(K)] \perp\!\!\!\perp A_i | \boldsymbol{x}_i \qquad \forall i \in [-I], \quad (2)$$

which means $[Y_i(1), \ldots, Y_i(K)]$ *are independent of* A_i *given* $\boldsymbol{x}_i$.

The above assumption holds in Example 1 when users' preferences of videos are observed. According to Table 2, in each subgroup of users with the same video preferences, their potential click rates are fixed and independent of the action we observe.

2.2 Model of the Online Decision Environment

In our online decision model, users arrive sequentially, and the decision maker selects an action from $[K]$ for each user. Formally, we consider a discrete time system $t \in [T]$, where $T \in \mathbb{N}_+$ and $[T] \triangleq \{1, \ldots, T\}$. In each time slot t, one and only one user arrives, and the user is associated with a context $\boldsymbol{x}_t \in \mathcal{X}$ and unobserved confounder $\boldsymbol{u}_t \in \mathcal{U}$. Then, the decision maker chooses an action a_t, and observes the outcome (or reward) y_t corresponding to it.

Consider that the confounders $(\boldsymbol{x}_t, \boldsymbol{u}_t), \forall t \in [T]$ are independently identically generated from a cumulative distribution function

$$F_{\boldsymbol{X},U}(\boldsymbol{x}, \boldsymbol{u}) \triangleq \mathbb{P}[\boldsymbol{X} \leq \boldsymbol{x}, \boldsymbol{U} \leq \boldsymbol{u}], \quad (3)$$

where $\boldsymbol{X} \in \mathcal{X}$ and $\boldsymbol{U} \in \mathcal{U}$ denote two random variables. The distribution $F_{\boldsymbol{X},U}(\boldsymbol{x}, \boldsymbol{u})$ characterizes the collective distribution of the confounders over the whole user population. If we marginalize over $\boldsymbol{u}$, then the observed confounder $\boldsymbol{x}_t$ is independently identically generated from the marginal distribution $F_{\boldsymbol{X}}(\boldsymbol{x}) \triangleq \mathbb{P}[\boldsymbol{X} \leq \boldsymbol{x}]$. Let the random variable $Y_t(k)$ denote the outcome of taking action k in time slot t. The following assumption captures that the outcome $Y_t(k)$ in time slot t is independent of the actions of other time slots.

Assumption 3 (Stable unit for online model). *The outcome in a time slot is independent of the actions in other time slots:*

$$\mathbb{P}[Y_t(k)=y | A_t=a_t, A_s=a_s] = \mathbb{P}[Y_t(k)=y | A_t=a_t], \forall y \in \mathcal{Y}, s \neq t \in [T]. \quad (4)$$

Furthermore, in the online setting, before the decision maker chooses the action, the "potential outcomes" $[Y_t(1), \ldots, Y_t(K)]$ are determined given the confounders $(\boldsymbol{x}_t, \boldsymbol{u}_t)$. Because the unobserved confounders $\boldsymbol{u}_t$ are i.i.d. in different time slots, the potential outcomes are independent of how we select the action, given the user's context $\boldsymbol{x}_t$. Formally, we have the following property.

Property 1. *The outcome for different actions in time slot t satisfies*

$$[Y_t(1), \ldots, Y_t(K)] \perp\!\!\!\perp A_t | \boldsymbol{x}_t \qquad \forall t \in [T]. \quad (5)$$

One can see that Assumption 1 and 2 for the logged data correspond to Assumption 3 and Property 1 for the online decision model. This way, we can *"connect"* the logged data with the online decision enviroment. Figure 2 summarizes our models of logged data and the online feedbacks. The offline data is orderless because of the stable unit Assumption 1 where each data samples are independent.

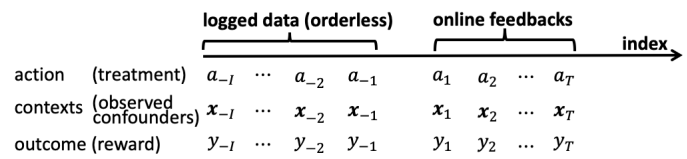

Figure 2: Summary of logged data and online feedbacks

2.3 The Online Decision Problems

The decision maker needs to select an action in each time slot. We consider the following two kinds of online decision problems based on whether users with different contexts are treated differently.

• **Context-independent decision problem.** Consider a company which makes a uniform decision for all incoming users. For example, for some user-interface, a uniform design for all users is preferred because it is easy to use and easy to develop. In causal inference, this context-independent setting corresponds to the estimation of "average treatment effect"[32]. In online learning, this setting corresponds to the "stochastic multi-armed bandit" problem[24]. When selecting action a_t, the decision maker can utilize the logged data $\mathcal{L}$ and the feedback history up to time slot t defined by

$$\mathcal{F}_t \triangleq \{(a_1, \boldsymbol{x}_1, y_1), \ldots, (a_{t-1}, \boldsymbol{x}_{t-1}, y_{t-1})\}.$$

Let $\mathcal{A}(\cdot, \cdot)$ denote an online decision algorithm, which outputs an action given logged data and feedback history. Formally, we have $a_t = \mathcal{A}(\mathcal{L}, \mathcal{F}_t)$. Define the best action as

$$a^* = \arg\max_{a \in [K]} \sum_{t=1}^{T} \mathbb{E}[y_t | a_t = a]. \quad (6)$$

We define the following context-independent pseudo-regret (abbreviated as *regret*) to measure the performance of algorithm $\mathcal{A}$

$$R(T, \mathcal{A}) \triangleq \sum_{t=1}^{T} \left(\mathbb{E}[y_t | a^*] - \mathbb{E}[y_t | a_t = \mathcal{A}(\mathcal{L}, \mathcal{F}_t)] \right). \quad (7)$$

A lower regret implies that the algorithm $\mathcal{A}$ achieves a higher context-independent reward. The decision maker's problem is to find a context-independent algorithm $\mathcal{A}$ to minimize its regret.

• **Context-dependent decision problem.** Consider that a company can make different decisions for different users. Formally, we model this by $a_t = \mathcal{A}_c(\mathcal{L}, \mathcal{F}_t, \boldsymbol{x}_t)$, where $\mathcal{A}_c$ denotes the context-dependent decision algorithm. Given the context $\boldsymbol{x}_t$, the *unknown* optimal reward is $\max_{a\in[K]} \mathbb{E}[y_t|a, \boldsymbol{x}_t]$. We define context-dependent pseudo-regret (abbreviated as *regret*) of algorithm $\mathcal{A}_c$ as $R(T, \mathcal{A}_c) \triangleq \sum_{t=1}^{T} \left(\max_{a\in[K]} \mathbb{E}[y_t|a, \boldsymbol{x}_t] - \mathbb{E}[y_t|a_t = \mathcal{A}_c(\mathcal{L}, \mathcal{H}_t, \boldsymbol{x}_t), \boldsymbol{x}_t] \right)$. Similarly, the decision maker's problem is to find a context-dependent algorithm $\mathcal{A}_c$ to minimize its regret.

3 GENERAL ALGORITHMIC FRAMEWORK

In this section, we develop a general algorithmic framework to utilize logged data to reduce the regret of online decision algorithms. We also prove its regret bounds, revealing the impacts of using logged data on online decision algorithms. We will study several typical algorithm instances under this framework in later sections.

3.1 The Design of the Algorithmic Framework

Figure 3 depicts the two components of our framework. The *"online bandit oracle"* models a general online learning algorithm. The *"offline evaluator"* models a general algorithm to synthesizes feedbacks from the logged data for the purpose of "training" the bandit oracle. Algorithm 1 outlines how to coordinate these two components to make decisions sequentially in T rounds. Each decision round contains an offline phase and an online phase. In the offline phase (Line 4 - Line 11), we use an *offline evaluator* to simulate outcomes from the logged data so to train the online bandit oracle. Formally, we first generate a random context according to the c.d.f. $F_{\boldsymbol{X}}(\cdot)$, and then we call the online bandit oracle and return a synthetic feedback to the oracle. We repeat such procedure until the offline evaluator cannot synthesize a feedback for this stage. Then the algorithm turns into the online phase (Line 12 - Line 14), where we call the same online bandit oracle to choose the action, and update it with the feedback of outcomes from the online environment.

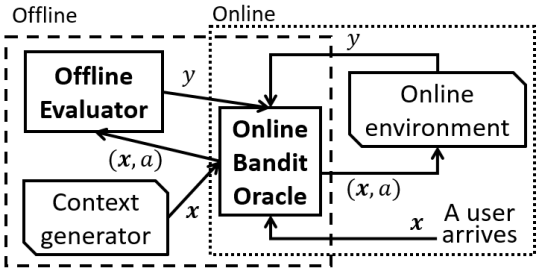

Figure 3: Illustration of our algorithmic framework

- **Online bandit oracle.** An online bandit oracle represents a general online learning algorithm. We consider a bandit oracle, which make sequential decisions in T' rounds. In each round $t \in [T']$, the bandit oracle receives a context $\boldsymbol{x}_t$, and then"plays" one action $a_t \in [K]$. Finally, the oracle "updates" itself with the feedback of the outcome y_t w.r.t. a_t. The outcome of other actions $a \neq a_t$ are not revealed in round t, as in the multi-armed "bandit" setting[6]. There are many instances of the online bandit oracle, including UCB[6], EXP3[7], Thompson sampling[2], LinUCB[27] and our ϵ-greedy causal forest algorithm (in Section 5), etc.
- **Offline evaluator.** The offline evaluator synthesizes outcomes as feedbacks using logged data so to "train" the online learning oracle. Given a context $\boldsymbol{x}$ and an action a, the offline evaluator searches the logged data and returns a "synthetic outcome" y. For example, if there is a data item $(a_i, \boldsymbol{x}_i, y_i)$ in $\mathcal{L}$ satisfying $a_i = a, \boldsymbol{x}_i = \boldsymbol{x}$, then the offline evaluator may return y_i and eliminates $(a_i, \boldsymbol{x}_i, y_i)$ from $\mathcal{L}$. Otherwise it returns NULL. We will introduce more offline evaluators later.

Algorithm 1: Algorithmic framework

```
1  Initialize the offline_evaluator with logged data L
2  Initialize the online_bandit_oracle
3  for t = 1 to T do
       // the offline phase
4      while True do
5          x ← context_generator() // based on F_X(·)
6          a ← online_bandit_oracle.play(x)
7          y ← offline_evaluator.get_outcome(x, a)
8          if y ≠ NULL then
9              online_bandit_oracle.update(x, a, y)
10         else
11             break
       // the online phase, t^th user comes with x_t
12     a_t ← online_bandit_oracle.play(x_t)
13     y_t ← the outcome from the online environment
14     online_bandit_oracle.update(x_t, a_t, y_t)
```

Unifying causal inference and online bandit learning. Both online bandit algorithms and causal inference algorithms are special cases of our framework. First, if there are no logged data, then the offline evaluator cannot synthesize feedbacks and will always return "NULL". In this case, our framework always calls the online bandit oracle, and reduces to an online bandit algorithm. Second, let's consider the case where T=1. Moreover, we consider a specific A/B test online learning oracle described as an objective-oriented class in Class 2. Then, after the offline phase, the estimated outcome $\bar{y}_a$ for action a can be used to estimate the causal effect. In this case, our framework reduces to a causal inference algorithm.

Class 2: Online Bandit Oracle - A/B test

Member variables: the average outcome $\bar{y}_a$ of each action $a \in [K]$, and the number of times n_a that action a was played.
Function play($\boldsymbol{x}$):
  return a with probability $1/K$ for each $a \in [K]$
Function update($\boldsymbol{x}, a, y$):
  $\bar{y}_a \leftarrow n_a \bar{y}_a/(n_a + 1)$, $n_a \leftarrow n_a + 1$

3.2 Regret Analysis Framework

We first decompose the regret of our framework Algorithm 1 as

regret of online plays = total regret - regret of virtual plays. (8)

The intuition is that among all the decisions made by the online bandit oracle, there are "virtual plays" whose feedbacks are simulated from *logged data* and "online plays" whose feedbacks are from *online environment*. The online bandit oracle cannot distinguish the "virtual plays" from "online plays". Thus we can further apply the theories of the online bandit oracles[6][1] to bound the total regret for both the "virtual plays" and "online plays". By subtracting the regret of the "virtual plays", we get the regret bounds for the "online plays". Following this idea, we obtain the following theorems.

Theorem 1 (general upper bound). *Suppose (1) for the online bandit oracle O, there exists a function $g(T)$, such that the regret of only using the online feedbacks $R(T,O)\leq g(T)$ $\forall T$; (2) the offline evaluator $\mathcal{M}$ returns unbiased outcomes, i.e. $\mathbb{E}[\mathcal{M}(\boldsymbol{x},a)]=\mathbb{E}[y|a]$ for the context-independent case, or $\mathbb{E}[\mathcal{M}(\boldsymbol{x},a)]=\mathbb{E}[y|a,\boldsymbol{x}]$ for the contextual case. Suppose the offline evaluator returns $\{\tilde{y}_j\}_{j=1}^N$ w.r.t. $\{(\tilde{\boldsymbol{x}}_j,\tilde{a}_j)\}_{j=1}^N$ till time T. Then, for a contextual-independent algorithm $\mathcal{A}$,*

$$R(T,\mathcal{A}) \leq g(T+N) - \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a] - \mathbb{E}[y|a=\tilde{a}_j]\right).$$

For a contextual algorithm $\mathcal{A}_c$, we also have a regret bound :

$$R(T,\mathcal{A}_c) \leq g(T+N) - \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a,\tilde{\boldsymbol{x}}_j] - \mathbb{E}[y|a=\tilde{a}_i,\tilde{\boldsymbol{x}}_i]\right).$$

Proof. All proofs are in the appendix. □

Note that the term $\max_{a\in[K]}\mathbb{E}[y|a]-\mathbb{E}[y|a=a_i]$ we subtract is the "regret" of the action in the i^{th} logged item compared to the optimal decision. Theorem 1 states that Algorithm 1 can reduce the regret, and it quantifies the reduction of regret by using the logged data.

Theorem 2 (general lower bound). *Suppose for any bandit oracle $\widetilde{O}$, $\exists$ a non-decreasing function $h(T)$, s.t. $R(T,\widetilde{O})\geq h(T)$ for $\forall T$. The offline estimator returns unbiased outcomes $\{\tilde{y}_j\}_{j=1}^N$ w.r.t. $\{(\tilde{\boldsymbol{x}}_j,\tilde{a}_j)\}_{j=1}^N$. Then for any contextual-independent algorithm $\widetilde{\mathcal{A}}$ we have:*

$$R(T,\widetilde{\mathcal{A}}) \geq h(T) - \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a] - \mathbb{E}[y|a=\tilde{a}_j]\right).$$

We also have a regret bound for any contextual algorithm $\widetilde{\mathcal{A}}_c$:

$$R(T,\widetilde{\mathcal{A}}_c) \geq h(T) - \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a,\tilde{\boldsymbol{x}}_j] - \mathbb{E}[y|a=\tilde{a}_j,\tilde{\boldsymbol{x}}_j]\right).$$

Theorem 2 states how we can apply the regret lower bound of online bandit oracles[12] to derive a regret lower bound with logged data. When an algorithm's upper bound meets the lower bound, we get a *nearly optimal* algorithm for decision making with logged data.

4 CASE STUDY I: CONTEXT-INDEPENDENT ONLINE DECISIONS

We apply the framework developed in Section 3 to study the context-independent decision setting. More specifically, we apply Algorithm 1 to speed up the UCB context-independent online algorithm with three instances of offline evaluator summarized in Table 6. We also derive refined regret upper bounds, revealing deeper insights on the power of logged data in speeding up online learning.

4.1 A Warm Up Case - Exact Matching

Algorithm description. Let us start with the instance $\mathcal{A}_1$ that applies Algorithm 1 to speed up the UCB algorithm with "exact matching" (i.e., a simple method in causal inference). The online bandit oracle Upper Confidence Bound (UCB) [6] is described as an objective-oriented Class 3. In each decision round, the oracle selects an action with the maximum upper confidence bound defined as $\bar{y}_a+\beta\sqrt{2\ln(n)/n_a}$, where $\bar{y}_a$ is the average outcome, β is a constant, and n_a is the number of times that an action a was played (Line 1).

Table 6: Instances of context-independent algorithms

	Instance $\mathcal{A}_1$	**Instance $\mathcal{A}_2$**	**Instance $\mathcal{A}_3$**
Offline evaluator	Exact matching	PS matching	IPS weighting
Online oracle	UCB	UCB	UCB

Class 3: Online Bandit Oracle - Upper Confidence Bound

1 **Member variables:** the average outcome $\bar{y}_a$ of each action $a\in[K]$, and the number of times n_a that action a was played.
2 **Function play**$(\boldsymbol{x})$**:**
3 **return** $\arg\max_{a\in[K]}\bar{y}_a+\beta\sqrt{\frac{2\ln(n)}{n_a}}$ // $(n=\sum_{a\in[K]}n_a)$
4 **Function update**$(\boldsymbol{x},a,y)$**:**
5 $\bar{y}_a\leftarrow(n_a\bar{y}_a+y)/(n_a+1)$, $n_a\leftarrow n_a+1$

Furthermore, we instantiate the offline evaluator with the "exact matching"[34] method outlined in Class 4. It searches for a data item in log $\mathcal{L}$ with the exact same context $\boldsymbol{x}$ and action a, and returns the outcome of that data item. If it cannot find a matched data item for an action a, then it stops the subsequent matching process for the action a. The stop of matching is to ensure that the synthetic feedbacks simulate the online feedbacks correctly.

Class 4: Offline Evaluator - Exact Matching

1 **Member variables**: $S_a\in\{False,True\}$ indicates whether we stop matching for action a, initially $S_a\leftarrow False$, $\forall a\in[K]$.
2 **Function get_outcome**$(\boldsymbol{x},a)$**:**
3 **if** $S_a=False$ **then**
4 $\mathcal{I}(\boldsymbol{x},a)\leftarrow\{i\mid\boldsymbol{x}_i=\boldsymbol{x},a_i=a\}$
5 **if** $\mathcal{I}(\boldsymbol{x},a)\neq\emptyset$ **then**
6 $i\leftarrow$ a random sample from $\mathcal{I}(\boldsymbol{x},a)$
7 $\mathcal{L}\leftarrow\mathcal{L}\backslash\{(a_i,\boldsymbol{x}_i,y_i)\}$
8 **return** y_i
9 $S_a\leftarrow True$
10 **return** NULL

Theorem 3 (Exact matching + UCB). *Assumptions 1, 2, 3 hold. Suppose there are C possible categories of users' features $\boldsymbol{x}^1,\ldots,\boldsymbol{x}^C$. Denote $\widehat{\mathbb{P}}[\boldsymbol{x}^c]$ as the fraction of online users whose context is $\boldsymbol{x}^c$, whose expectation is $\mathbb{P}[\boldsymbol{x}^c]\triangleq\mathbb{E}[\widehat{\mathbb{P}}[\boldsymbol{x}^c]]$. Denote $a^*\triangleq\arg\max_{\tilde{a}\in[K]}\mathbb{E}[y|\tilde{a}]$, $\Delta_a\triangleq\mathbb{E}[y|a^*]-\mathbb{E}[y|$ Let $N(\boldsymbol{x}^c,a)\triangleq\sum_{i\in[-I]}\mathbb{1}_{\{\boldsymbol{x}_i=\boldsymbol{x}^c,a_i=a\}}$ be the number of samples with context $\boldsymbol{x}^c$ and action a. Then,*

$$R(T,\mathcal{A}_1)\leq\sum_{a\neq a^*}\left(1+\frac{\pi^2}{3}+\sum_{c\in[C]}\max\left\{0,8\frac{\ln(T+A)}{\Delta_a^2}\widehat{\mathbb{P}}[\boldsymbol{x}^c]-\min_{\tilde{c}\in[C]}\frac{N(\boldsymbol{x}^{\tilde{c}},a)\mathbb{P}[\boldsymbol{x}^c]}{\mathbb{P}[\boldsymbol{x}^{\tilde{c}}]}\right\}\right)\Delta_a,$$

where the constant

$$A=N-\sum_{a\neq a^*}\sum_{c\in[C]}\max\left\{0,\ N(\boldsymbol{x}^c,a)-(8\frac{\ln(T+N)}{\Delta_a^2}+1+\frac{\pi^2}{3})\mathbb{P}[\boldsymbol{x}^c]\right\}.$$

Theorem 3 states how logged data reduces the regret. When there is no logged data, i.e. $N(\boldsymbol{x}_c, a) = 0$, the regret bound $O(\log(T))$ is the same as that of UCB. If the number of logged data $N(\boldsymbol{x}^c, a)$ is greater than $\widehat{\mathbb{P}}[\boldsymbol{x}^c]8\ln(T+A)/\Delta_a^2$ for each context $\boldsymbol{x}^c$ and action a, then the regret is smaller than a constant. In addition, when the regret Δ_a of choosing an action $a \neq a^*$ is smaller, we need more logged data to make the regret close to zero.

We point out that "Historical UCB" (HUCB) algorithm[33] is a special case of our algorithm $\mathcal{A}_1$. Because HUCB ignores the context, we consider a dummy context $\boldsymbol{x}^1$. Then, we have $\widehat{\mathbb{P}}_a[\boldsymbol{x}^1]=1$ for $\forall a \in [K]$, and our regret bound is similar to that of HUCB.

One limitation of the exact matching evaluator is that when $\boldsymbol{x}$ is continous or has a high dimension, we can hardly find a sample in log-data with exactly the same context. To address this limitation, we next consider the propensity score matching method[34]

4.2 Improving Matching Efficiency via Propensity Score Matching

We instantiate the offline evaluator of Algorithm 1 with propensity score matching outlined in Class 5. Together with the UCB oracle (Class 3), we get algorithm $\mathcal{A}_2$. The propensity score $p_i \in [0,1]$ is the probability of observing the action a_i given the context $\boldsymbol{x}$, i.e. $p_i = \mathbb{P}[a_i|\boldsymbol{x}_i]$. For the context-independent case, the Assumption 2 implies the ignorability property given the *propensity score*[31], i.e.

$$[Y_i(1), \ldots, Y_i(K)] \perp\!\!\!\perp A_i | p_i, \quad \text{where } p_i = \mathbb{P}[A_i = a_i | \boldsymbol{x}_i]. \tag{9}$$

That is why one only needs to match the one-dimensional propensity score instead of all the contexts. In Class 5, we replace the full context $\boldsymbol{x}$ in Class 4 to the propensity score p. We use the strategy of stratification[8], where we round the propensity scores to be their nearest pivots in set Q and use the rounded values to do matching.

Class 5: Offline Evaluator - Propensity Score Matching

1 **Member variables**: $S_a \in \{False, True\}$ indicates whether we stop matching for action a, initially $S_a \leftarrow False, \forall a \in [K]$. The pivot set $Q \subset [0,1]$ has finite elements
2 **Function get_outcome**(p, a)**:**
3 **if** $S_a = False$ **then**
4 $\mathcal{I}(p,a) \leftarrow \{i \mid \texttt{stratify}(p_i) = \texttt{stratify}(p), a_i = a\}$
5 **if** $\mathcal{I}(p,a) \neq \emptyset$ **then**
6 $i \leftarrow$ a random sample from $\mathcal{I}(p,a)$
7 $\mathcal{L} \leftarrow \mathcal{L} \backslash \{(a_i, \boldsymbol{x}_i, y_i)\}$
8 **return** y_i
9 $S_a \leftarrow True$
10 **return** NULL
11 **Function stratify**(p)**:**
12 **return** $\arg\min_{q \in Q} |p - q|$ `// use the rounded value`

Propensity scores are widely used in offline causal inference[8] and offline policy evaluation[25][35]. The propensity score can be recorded by the company's logger[27]. The propensity score can also be predicted from the logged data via machine learning[28].

Theorem 4 (Propensity score matching + UCB). *Assumptions 1, 2, 3 hold. We consider that the propensity scores are in a finite set* $p_i \in Q \triangleq \{q_1, \ldots, q_Q\} \subseteq [0,1]$ *for* $\forall i \in [-I]$. *Let* $N(q,a)$ *be the number of data items whose propensity score* $p_i = q$ *and action* $a_i = a$. *Denote* $\widehat{\mathbb{P}}[q_c]$ *as the fraction of online users whose propensity score is* q_c. *Then,*

$$R(T, \mathcal{A}_2) \leq \sum_{a \neq a^*} \left(1 + \frac{\pi^2}{3} + \sum_{c \in [Q]} \max\left\{ 0, 8\frac{\ln(T+A)}{\Delta_a^2} \widehat{\mathbb{P}}[q_c] - \min_{\tilde{c} \in [Q]} \frac{N(q_{\tilde{c}}, a)\mathbb{P}[q_c]}{\mathbb{P}[q_{\tilde{c}}]} \right\} \right) \Delta_a,$$

where $\mathbb{P}[q_c] \triangleq \mathbb{E}[\widehat{\mathbb{P}}[q_c]]$ *is the probability for propensity score* q_c, *and*

$$A = N - \sum_{a \neq a^*} \sum_{c \in [Q]} \max\left\{ 0, \min_{\tilde{c} \in [Q]} \frac{N(q_{\tilde{c}}, a)\mathbb{P}[q_c]}{\mathbb{P}[q_{\tilde{c}}]} - (8\frac{\ln(T+N)}{\Delta_a^2} + 1 + \frac{\pi^2}{3})\mathbb{P}[q_c] \right\}.$$

The regret bound is similar to that of $\mathcal{A}_1$ (exact-matching). But propensity score matching increases the chances to find a matched data sample since now we only need to find a sample with matched propensity score, and thus algorithm $\mathcal{A}_2$ further reduce the regret.

4.3 Weighting Methods in Our framework

To further demonstrate the applicability of our framework, we now show how to use weighting methods[35][23] in causal inference.

Inverse propensity score weighting. We use the inverse of the propensity score $1/p$ as the weight, and the estimated outcome $\bar{y}_a$ is a weighted average from data (Line 4). The intuition of weighting by the inverse propensity score is as follows: if an action is applied to a group A of users more often than other groups of users, then each sample for group A should have smaller weight so that the total weights of each group is proportional to its population. In fact, the IPS weighting estimator is unbiased via *importance sampling*[32].

Class 6 shows the inverse propensity score weighting evaluator. We first estimate the outcome $\bar{y}_a$ as the weighted average of data items with action a. Then, we calculate the effective number N_a of logged plays of action a, based on Hoeffding's inequalities[22]. After such initialization, the offline evaluator will return $\bar{y}_a$ w.r.t. the input action a for $\lfloor N_a \rfloor$ times, and return NULL afterwards.

Class 6: Offline Evaluator - IPS Weighting

1 **Member variables:** $\bar{y}_a, N_a (a \in [K])$ initialized in **__init__**$(\mathcal{L})$
2 **Function __init__**$(\mathcal{L})$**:**
3 **for** $a \in [K]$ **do**
4 $\bar{y}_a \leftarrow \frac{\sum_{i \in [-I], a_i = a} y_i / p_i}{\sum_{i \in [-I], a_i = a} 1/p_i}$ and $N_a \leftarrow \frac{(\sum_{i \in [-I], a_i = a} 1/p_i)^2}{\sum_{i \in [-I], a_i = a} (1/p_i)^2}$
5 **Function get_outcome**$(\boldsymbol{x}, a)$**:**
6 **if** $N_a \geq 1$ **then**
7 $N_a \leftarrow N_a - 1$
8 **return** $\bar{y}_a$
9 **return** NULL

Theorem 5 (Inverse propensity score weighting + UCB). *Assumptions 1, 2, 3 hold. Suppose the online reward is bounded* $y_t \in [0,1]$ $\forall t \in [T]$ *and the propensity score is bounded* $p_i \geq \bar{s} > 0$ *for* $\forall i \in [I]$. *Then*

$$R(T, \mathcal{A}_3) \leq \sum_{a \neq a^*} \Delta_a \left(\max\left\{ 0, 8\frac{\ln(T + \sum_{a=1}^K \lceil N_a \rceil)}{\Delta_a^2} - \lfloor N_a \rfloor \right\} + (1 + \frac{\pi^2}{3}) \right)$$

where we recall $N_a = \left(\sum_{i \in [-I]} \frac{1}{p_i} \mathbb{1}_{\{a_i = a\}} \right)^2 / \sum_{i \in [-I]} \left(\frac{1}{p_i} \mathbb{1}_{\{a_i = a\}} \right)^2$.

Theorem 5 quantifies the impact of the logged data on the regret of the algorithm $\mathcal{A}_3$. Recall that N_a is the equivalent number of

feedbacks for an action a. When there is no logged data, i.e. $N_a = 0$, the regret bound reduces to the $O(\log T)$ bound of UCB. A larger N_a indicates a lower regret bound (or a higher reduction of regret). Notice that the number N_a depends on the distribution of logged data items' propensity scores. In particular, when all the propensity scores are a constant $\tilde{p}$, i.e. p_i=$\tilde{p}$ $\forall i$, the effective number is the actual number of plays of action a, i.e. N_a=$\sum_{i\in[-I]} \mathbb{1}_{\{a_i=a\}}$. When the propensity scores $\{p_i\}_{i\in[-I]}$ have a more skewed distribution, the number N_a will be smaller, leading to a larger regret bound.

Discussions. Note that our framework is not limited to the above three instances. One can replace the online bandit oracle to A/B testing, ϵ-greedy[24], EXP3[7] or Thompson sampling[2]. One can do weighting via other methods such as the balanced weighting[23]. One can also use supervised learning[40] or apply techniques such as "doubly robust"[17] to construct the offline evaluator.

4.4 Dropping the Ignorability Assumption

All the above theorems rely on the ignorability Assumption 2. We now analyze the algorithms when this assumption does not hold.

To see the impact of dropping Assumption 2, consider Example 1. But now, the logged data do not record *users' preferences to video*. Then our *causal inference* Strategy 2 will do the same calculation as the *empirical mean* Strategy 1 which selects the wrong action.

Theorem 6 (No ignorability). *Assumptions 1, 3 hold. For a context-independent algorithm $\mathcal{A}$ using the UCB oracle, suppose the offline evaluator returns $\{y_j\}_{j=1}^N$ w.r.t. $\{(\boldsymbol{x}_j,a_j)\}_{j=1}^N$. The bias of the average outcome for action a is $\delta_a \triangleq (\sum_{j=1}^N \mathbb{1}_{\{a_j=a\}} y_j)/(\sum_{j=1}^N \mathbb{1}_{\{a_j=a\}}) - \mathbb{E}[y|a]$. Denote $N_a \triangleq \sum_{j=1}^N \mathbb{1}_{\{a_j=a\}}$, and recall a^* is the optimal action. Then,*

$$R(T,\mathcal{A}) \le \sum_{a\neq a^*} \Delta_a \left(16\frac{\ln(N_a+T)}{\Delta_a^2} - 2N_a\left(1-\frac{\max\{0,\,\delta_a-\delta_{a^*}\}}{\Delta_a}\right)+\left(1+\frac{\pi^2}{3}\right)\right).$$

Theorem 6 states the relationship between the bias of the offline evaluator (i.e. δ_a) and the algorithm's regret. When the ignorability holds, δ_a=0 $\forall a\in[K]$ for all the above algorithms $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$. In this case, the bound in Theorem 6 is similar to the previous bounds in Theorem 3, 4 and 5 except that we raise the constant from 8 to 16. As $\delta_a - \delta_{a^*}$ increases, i.e. the offline evaluator has a greater bias for an inferior action than the bias of the optimal action, then the regret upper bound increases. In Theorem 6, we also notice a sufficient condition to reduce the regret upper bound by using logged data is $1-\max\{0, \delta_a-\delta_{a^*}\}/\Delta_a > 0$, or, $\delta_a - \delta_{a^*} < \Delta_a$ for $\forall a \neq a^*$. The physical meaning is that when the estimated outcome of the optimal action is greater than that of other actions, the logged data help to find the optimal action, and thus reduce the regret.

5 CASE STUDY II: CONTEXTUAL DECISIONS

We apply the framework developed in Section 3 to study the contextual decision making problem. More specifically, we apply Algorithm 1 to speed up the LinUCB and causal forest, which represent a parametric and non-parametric online algorithms respectively. Table 7 summarizes the algorithm instances in this section.

Table 7: Instances for the contextual-dependent algorithms

	Instance $\mathcal{A}_4$	**Instance $\mathcal{A}_5$**
Offline evaluator	Linear regression	Matching on forest
Online bandit oracle	LinUCB	ϵ-greedy-causal-forest

5.1 Contextual Decision with Linear Reward

We first consider that the outcomes (or rewards) have a linear form:

$$y_t = \boldsymbol{\theta}\cdot\phi(\boldsymbol{x}_t, a_t) + \epsilon \qquad \forall t \in [T], \tag{10}$$

where $\phi(\boldsymbol{x}, a) \in \mathbb{R}^m$ is an m-dimensional *known* feature vector. The $\boldsymbol{\theta}$ is an m-dimensional *unknown* parameter to be learned, and ϵ is a random noise with $\mathbb{E}[\epsilon]$=0. We instantiate Algorithm 1 with "LinUCB" (Class 7) as the online bandit oracle and "linear regression" (Class 8) as the offline evaluator, to get an algorithm instance $\mathcal{A}_4$.

LinUCB oracle. We use the Linear Upper Confidence Bound (LinUCB) algorithm[27] in Class 7 as the online learning oracle. The oracle estimates the unknown parameter $\hat{\boldsymbol{\theta}}$ based on the feedbacks. The $\hat{y}_a \triangleq \hat{\boldsymbol{\theta}}^T\phi(\boldsymbol{x},a) + \beta_t\sqrt{\phi(\boldsymbol{x},a)^T V^{-1}\phi(\boldsymbol{x},a)}$ is the upper confidence bound of reward, where $\{\beta_t\}_{t=1}^T$ are parameters. The oracle always plays the action with the highest upper confidence bound.

Class 7: Online Bandit Oracle - LinUCB

1 **Member variables:** a matrix V (initially V is a $d\times d$ matrix), a d-dimensional vector $\boldsymbol{b}$ (initially $\boldsymbol{b}$=$\boldsymbol{0}$ is zero), time t=1 initially
2 **Function play**$(\boldsymbol{x})$**:**
3 $\hat{\boldsymbol{\theta}} \leftarrow V^{-1}\boldsymbol{b}$
4 **for** $a \in [K]$ **do**
5 $\hat{y}_a \leftarrow \hat{\boldsymbol{\theta}}^T\phi(\boldsymbol{x},\boldsymbol{a}) + \beta_t\sqrt{\phi(\boldsymbol{x},a)^T V^{-1}\phi(\boldsymbol{x},a)}$
6 **return** $\arg\max_{a\in[K]} \hat{y}_a$
7 **Function update**$(\boldsymbol{x}, a, y)$**:**
8 $V \leftarrow V + \phi(\boldsymbol{x},a)\phi(\boldsymbol{x},a)^T, \quad \boldsymbol{b} \leftarrow \boldsymbol{b} + y\boldsymbol{x}, \quad t \leftarrow t+1$

Linear regression offline evaluator. Class 8 shows how we use linear regression to construct the offline evaluator. From the logged data, we estimate the parameter $\hat{V}$ (Line 3), and the parameter $\hat{\boldsymbol{\theta}}$ (Line 4). The offline evaluator always return the estimated outcome $\phi(\boldsymbol{x},a)\cdot\hat{\boldsymbol{\theta}}$ by an linear model. The offline evaluator will stop returning outcomes, when the logged data cannot provide a tighter confidence bound than that of the online bandit oracle (Line 6 - 9).

Class 8: Offline Evaluator - Linear Regression

1 **Member variables:** $V, \hat{V}$ are $m\times m$ matriices, where $V/\hat{V}$ is for the online/offline confidence bounds. $\hat{\boldsymbol{\theta}}$ is the estimated parameters. The V is a shared with LinUCB oracle.
2 **Function** __**init**__$(\mathcal{L})$**:**
3 $\hat{V} \leftarrow \boldsymbol{I}_m + \sum_{i=1}^N \phi(\boldsymbol{x}_i,a_i)\cdot\phi(\boldsymbol{x}_i,a_i)^T$,
  `// I_m is m×m identity matrix`
4 $\boldsymbol{b} \leftarrow \sum_{i=1}^N y_i\cdot\phi(\boldsymbol{x}_i,a_i)$, $\hat{\boldsymbol{\theta}} \leftarrow \hat{V}^{-1}\boldsymbol{b}$
5 **Function get_outcome**$(\boldsymbol{x}, a)$**:**
6 **if** $||\phi(\boldsymbol{x},a)||_{V+\phi(\boldsymbol{x}_i,a_i)\cdot\phi(\boldsymbol{x}_i,a_i)^T} > ||\phi(\boldsymbol{x},a)||_{\hat{V}}$ **then**
7 $V \leftarrow V + \phi(\boldsymbol{x}_i,a_i)\cdot\phi(\boldsymbol{x}_i,a_i)^T$
8 **return** $\phi(\boldsymbol{x},a)\cdot\hat{\boldsymbol{\theta}}$
9 **return** NULL

We now show a regret bound for the problem-dependent case. Suppose in the specific problem, for any context $\boldsymbol{x}_t$, the difference of expected rewards between the best and the *"second best"* actions is at least $\Delta_{\min}$. This is the settings of section 5.2 in the paper[1].

Theorem 7 (Linear regression + LinUCB, problem dependent). *Assumptions 1, 2, 3 hold. In addition, the rewards satisfy the linear model in (10). Suppose offline evaluator returns a sequence* $\{y_i\}_{i=1}^N$ *w.r.t.* $\{(\boldsymbol{x}_i, a_i)\}_{i=1}^N$. *Let* $V_N \triangleq \sum_{i\in[N]} \boldsymbol{x}_i \boldsymbol{x}_i^T$, $L \triangleq \max_{t\leq T}\{||\boldsymbol{x}_t||_2\}$. *Then*

$$R(T, \mathcal{A}_4) \leq \frac{8d(1+2\ln(T))}{\Delta_{\min}} d\log(1+\kappa) + 1,$$

where $\kappa = TL^2/\lambda_{\min}(V_N)$. *In particular, when the smallest eigenvalue* $\lambda_{\min}(V_N) \geq (1/2 + \ln(T))TL^2$, *the regret is bounded by* $16d^2/\Delta_{\min}+1$.

In Theorem 7 we see for a fixed κ, the regret is $\log(T)$ in T time slots. Moreover, the above theorem highlights that when the logged data contains enough information, so that $\lambda_{\min}(V_N)$ is greater than $(1/2 + \ln(T))T$, the regret can be upper bounded by a constant.

5.2 Forest-based Online Decision Making

We now introduce the (non-parametric) forest-based algorithms for the contextual decision problem with logged data. We first propose an online bandit oracle based on the causal forest estimator which was proved to be unbiased and asymptotically normal[37][4].

ϵ-greedy causal forest oracle. A causal forest $C\mathcal{F}$ is a set of B decision trees. Each context $\boldsymbol{x}$ belongs to a leaf $L_b(\boldsymbol{x})$ in the tree $b \in [B]$. Given a set of data $\mathcal{D}=\{(\boldsymbol{x}_i, a_i, y_i)\}_{i=1}^D$, we can use a tree b to estimate the outcome of a context $\boldsymbol{x}$ and an action a as

$$\hat{L}_b(\boldsymbol{x}, a) \triangleq \frac{\sum_{i\in[D]} \mathbb{1}\{L_b(\boldsymbol{x}_i)=L_b(\boldsymbol{x})\}\mathbb{1}\{a_i=a\} y_i}{\sum_{i\in[D]} \mathbb{1}\{L_b(\boldsymbol{x}_i)=L_b(\boldsymbol{x})\}\mathbb{1}\{a_i=a\}}. \tag{11}$$

Class 9 describes the ϵ-greedy causal forest algorithm. For a context $\boldsymbol{x}$, the algorithm first estimates the outcome as the average in all trees (Line 4). Then, with probability $1-\epsilon_t$ in time t, the algorithm chooses the action with the largest estimated outcome. With probability ϵ_t, the algorithm randomly selects an action in order to explore its outcome. The parameter ϵ_t decreases and converges to 0 as $t \to +\infty$. The oracle will update the data $\mathcal{D}$ upon receiving a feedback (Line 7). It will also update the forest $C\mathcal{F}$ using the training algorithm `train_causal_forest` in the papers[37][4] (Line 8).

Class 9: Online Bandit Oracle - ϵ-Greedy Causal Forest

1 **Member variables:** the causal forest $C\mathcal{F}$ of B trees, data $\mathcal{D}$ with initial value $\emptyset$, t with initial value 1
2 **Function play($\boldsymbol{x}$):**
3 **for** $a \in [K]$ **do**
4 $\hat{y}_a \leftarrow \frac{1}{B}\sum_{b\in[B]} \hat{L}_b(\boldsymbol{x}, a)$
5 $a_t \leftarrow \begin{cases} \arg\max_{a\in[K]} \hat{y}_a & \text{with prob. } 1-\epsilon_t, \\ \text{a random action in } [K] & \text{with prob. } \epsilon_t. \end{cases}$
6 **Function update($\boldsymbol{x}, a, y$):**
7 $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{x}, a, y)\}$ and $t \leftarrow t+1$
8 $C\mathcal{F} \leftarrow$ `train_causal_forest`($\mathcal{D}$)

Theorem 8 (ϵ-greedy-causal-forest, asymptotic). *Assumptions 1, 2, 3 hold. The logged data satisfies the conditions of Theorem 1 in paper[37]. In particular, the tree-predictors are α-regular, i.e. "each split leaves at least a fraction $\alpha \leq 0.5$ of the available training examples on each side." When the exploration probability $\epsilon_t = t^{-1/2(1-\beta)}$, with no logged data, the asymptotic regret (for any small $\xi > 0$)*

$$\lim_{T\to+\infty} R(T, \mathcal{A}_5)/T^{(2+\xi)/(3-\beta)} < +\infty$$

where $\beta = 1 - \frac{2A}{2+3A}$ *and* $A = \frac{\pi}{d}\frac{\log((1-\alpha)^{-1})}{\log(\alpha^{-1})}$. *Hence* $\lim_{T\to+\infty} R(T, \mathcal{A}_5)/T = 0$.

Theorem 8 states that our ϵ-greedy causal forest algorithm achieves an online regret sub-linear to T. Note that the causal forest can be biased. We see by appropriate choices of the exploration rate ϵ_t, the above algorithm balances both the bias-variance tradeoff and the exploration-exploitation tradeoffs. We see that the lower bound of β increases as α decreases (or as dimensions d increases). The physical meaning is that if users' context vectors are distributed uniformly in the feature space (when α=0.5), then the asymptotic regret upper bound $T^{\beta_{\min}}$ reaches the minimal where $\beta_{\min}$ is $1-(1+d/\pi)^{-1}$.

Matching-on-forest offline evaluator. Class 10 describes the matching-on-forest offline evaluator. The idea is to find a weighted "nearest neighbor" in the logged data for the context-action pair $(\boldsymbol{x}, a)$ according to the decision trees. On a decision tree $b \in [B]$, the "nearest neighbors" of $(\boldsymbol{x}, a)$ is the data samples that are in the same leaf $L_b(\boldsymbol{x})$ and have the same action a. On other decision trees, $(\boldsymbol{x}, a)$ will have other "nearest neighbors". Therefore, we randomly pick one of the B trees (Line 3), and return one of its "nearest neighbors" on this tree (Line 4-6). If a data sample belongs to the nearest neighbors of more trees, then it will be returned more often.

Class 10: Offline Evaluator - Matching on Forest

1 **Input:** a causal forest $C\mathcal{F}$ with leaf functions $\{L_b\}_{b=1}^B$
2 **Function get_outcome($\boldsymbol{x}, a$):**
3 $b \leftarrow$ a uniformly random number in $[B]$
4 $\mathcal{I} \leftarrow \{i \mid L_b(\boldsymbol{x}_i) = L_b(\boldsymbol{x}), a_i = a\}$
5 **if** $\mathcal{I} \neq \emptyset$ **then**
6 $i \leftarrow$ a random sample from $\mathcal{I}$
7 $\mathcal{L} \leftarrow \mathcal{L} \backslash \{(a_i, \boldsymbol{x}_i, y_i)\}$
8 **return** y_i
9 **return** NULL

6 EXPERIMENTS

In this section, we evaluate our algorithms on both synthetic data. We compare the performance of three variants of our algorithm $\mathcal{A}$: (1) online bandit algorithm that only uses online feedbacks $\mathcal{A}^{\text{on}}$; (2) offline causal inference algorithm that only uses offline logged data $\mathcal{A}^{\text{off}}$; (3) the full algorithm $\mathcal{A}$ that use both data sources. We focus on demonstrating the benefits of unifying offline causal inference and online bandit algorithms. We do not intend to compare the performances of different offline evaluators or online algorithms and they are by no means exhaustive.

6.1 Synthetic data

Data generating process. We first generate a user's context $\boldsymbol{x}$ according to a d-dimensional uniform distribution. Second, we sample the propensity score p from $\boldsymbol{Q}=\{0.2, 0.4, 0.6, 0.8\}$. Third, we generate the action $a \in \{0, 1\}$ according to the propensity score, i.e. $\mathbb{P}[a = 0|\boldsymbol{x}]=p$. Fourth, we generate the outcome according to a function $y=f(\boldsymbol{x}, a)$. For the contextual-independent cases, the expected reward for an action a is $\mathbb{E}[y|a]=\mathbb{E}_{\boldsymbol{x}}[f(\boldsymbol{x}, a)|a]$ by marginalizing over $\boldsymbol{x}$. Detailed settings can be found in our technical report[3].

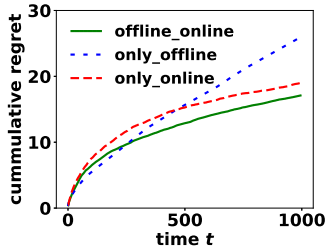

Figure 4: Regret of $\mathcal{A}_1$ (exact matching + UCB)

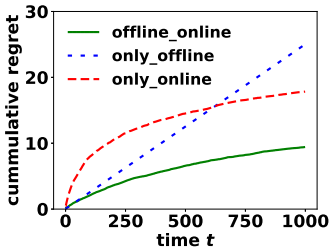

Figure 5: Regret of $\mathcal{A}_2$ (propensity score matching + UCB)

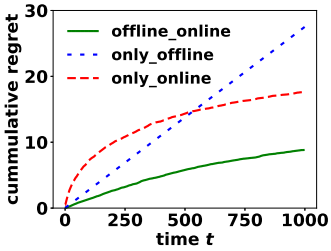

Figure 6: Regret of $\mathcal{A}_3$ (IPS weighting + UCB)

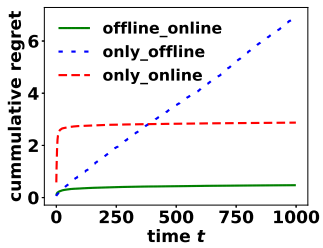

Figure 7: Regret of $\mathcal{A}_4$ (linear regr. + LinUCB), linear f

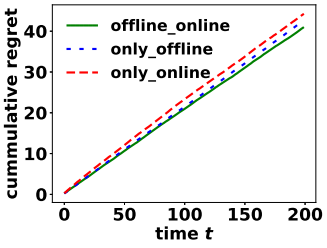

Figure 8: Regret of $\mathcal{A}_4$ (LinUCB), non-linear $\tilde{f}$

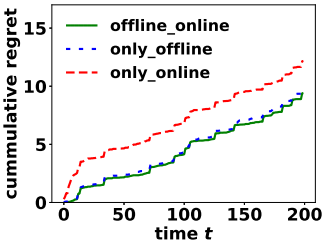

Figure 9: Regret of $\mathcal{A}_5$ (ϵ-greedy causal forest), non-linear $\tilde{f}$

Figure 10: The impact of the bias and the number of logged samples on the total regrets in 500 rounds, for $\mathcal{A}_3$

Evaluating all algorithms. We first evaluate context-independent algorithms, where we have 50 logged data points and 1000 online rounds. There are two actions with exptected rewards 0 and 0.5. We run each algorithm 200 times to get the average regret. Figure 4 shows the cummulative regrets of three variants of algorithm $\mathcal{A}_1$ that uses exact matching. First, we see the "only_offline" variant $\mathcal{A}_1^{\text{off}}$ has the highest total regret, and the regret increases linearly in time. This is because if we only use the logged data and make a bad decision, the bad decsion will persist and in the long run yields a high regret. Second, we observe our "offline_online" variant $\mathcal{A}_1$ is always better than the "only_online" $\mathcal{A}_1^{\text{on}}$. This is because using logged data to warm-start reduces the cost of online exploration, as proven by Theorem 3 in Section 4. Third, we notice that when t<200, the "offline_online" variant $\mathcal{A}_1$ has a slightly higher regret than that of "only_offline" $\mathcal{A}_1^{\text{off}}$. This is because $\mathcal{A}_1$ will initially *"explore"* the inferior actions. Figure 5 and Figure 6 show the regrets for three variants of $\mathcal{A}_2$ (propensity matching) and $\mathcal{A}_3$ (IPS weighting) respectively. We have similar observations. But the regret reduction w.r.t. online bandit algorithm $R(\mathcal{A}_2^{\text{on}},T)-R(\mathcal{A}_2,T)$ (or $R(\mathcal{A}_3^{\text{on}},T)-R(\mathcal{A}_3,T)$) is greater than $R(\mathcal{A}_1^{\text{on}},T)-R(\mathcal{A}_1,T)$ of $\mathcal{A}_1$. This is because through propensity score matching or weighting, the algorithm can match more data points and therefore reduce more regret, as indicated by Theorems 4 and 5 in Section 4.

We now investigate the contextual decision case. We evaluate three variants of $\mathcal{A}_4$ in Figure 7. The outcome function $y = f(\boldsymbol{x},a) = \boldsymbol{\theta}\cdot\phi(\boldsymbol{x},a)$ is linear which is parameterized by $\boldsymbol{\theta}$, where ϕ is a known function. We have similar observations with the context-independent case. First, the regret of the "only_offline" variant $\mathcal{A}_4^{\text{off}}$ increases linearly in time. Second, the regret of "only_online" $\mathcal{A}_4^{\text{on}}$ algorithm (i.e. LinUCB) stops increasing after the convergence. Third, the "offline_online" $\mathcal{A}_4$ has the lowest regret which is near zero. Notice that the regret of $\mathcal{A}_4$ is lower than that of $\mathcal{A}_4^{\text{on}}$ because it uses the logged data to save the cost of online exploration. In Figure 8, we evaluate $\mathcal{A}_4$ over a non-linear function $\tilde{f}(\boldsymbol{x},a)\triangleq(\sum_{j=1}^{d}\mathbb{1}_{\{\phi_j(\boldsymbol{x},a)\geq\theta_j\}})/d + 0.5\mathbb{1}_{\{a=1\}}$. We see all variants of $\mathcal{A}_4$ perform badly, and the regrets increase linearly in t. This is because the linear model is inherently biased as the linearity does not hold. In contrast, in Figure 9 we see when we use the non-parametric ϵ-greedy causal forest, i.e. $\mathcal{A}_5$, we can reduce the regrets of $\mathcal{A}_4$ by over 75% (from around 40 to less than 10).

Impact of the quantity and quality of logged data. In the ideal case, in terms of quantity we have a sufficiently large number of data for each action, and in terms of quality the data records all the confounding factors. In reality, these conditions may not hold.

In Figure 10, we investigate the impacts of both the quantity and quality of data, where we focus on the context-indepedent algorithm $\mathcal{A}_3$. Recall that the expected rewards for the two actions are 0 and 0.5. Now, in the logged data we add a bias to the first action, and its expected reward becomes "0+bias". We observe that when the bias is 0 or 0.3, the "offline_online" variant $\mathcal{A}_3$ has the lowest regret. This is because with small bias, the logged data is still informative to select the better action. However, when the bias is as large as 0.9, the "only_online" variant (i.e. UCB) achieves the lowest regret, because the offline estimations are misleading. The impact of the number of logged samples depends on the bias. In the case of zero bias (the left figure), if we have a large number of logged samples (e.g. 100), then the algorithms $\mathcal{A}_3$ and $\mathcal{A}_3^{\text{off}}$ have low regrets because they use logged data. But when logged data has high bias (the right figure), more logged samples result in a higher regret for algorithms $\mathcal{A}_3$ and $\mathcal{A}_3^{\text{off}}$ that use the logged data.

Lessons learned. Our "offline_online" algorithm that uses both the logged data and online feedbacks achieves the lowest regret. Using propensity scores can reduce the regret compared to the exact matching. When the data generating process does not satisfy the linear relationship, the LinUCB algorithm has a high regret, but ϵ-greedy causal-forest algorithm can reduce the regret.

7 RELATED WORKS

Offline causal inference[32][34][29] focuses on the counterfactual reasoning question "what will the outcome be if we had done another action before?" Pearl formulated a Structural Causal Model (SCM) framework to model and infer causal effects[29]. Rubin proposed another Potential Outcome (PO) framework[32]. Matching [28][34] and weighting [8][23][21] are important techniques that deal with the imblalance of action's distributions in offline data. Other important techniques include "doubly robust"[17] that combines regression and causal inference, and "differences-in-differences" [10] that uses data with timestamps. Many previous works considered the average treatment effect on the entire population. Recently, several works studied the individualized treatment effects[37][4]. For applications, people were re-thinking the recommendation problem as a causal inference problem[38]. Bottou et al.[11] applied causal inference to computational advertising. Offline policy evaluation is closely related to offline causal inference. It considers the performance of a policy that generates actions given contexts[35][25]. We also use offline policy evaluation to evaluate the performances of contextual bandit algorithms[26]. Our paper differs from the above works in that we consider the sequential online decisions after the offline causal inference. In particular, all the above algorithms can be seen as special cases in our algorithmic framework.

The multi-armed bandit (MAB) problem considers a reward-maximizing player who sequencially makes decisions and receives rewards as feedbacks. There is a tradeoff between "exploiting the empirically optimal decision" and "exploraing other potentially optimal decisions"[5]. From a frequentist's view, people proposed the UCB algorithm that chooses the arm with the highest upper confidence bound[6], and parametric variants such as LinUCB[14]. EXP3 algorithm[7] deals with the non-stochastic enviroment. For the contextual bandit problem, LinUCB algorithm was proved a $O(\sqrt{T\log(T)})$ regret bound[13][1] and could be applied to news article recommendation[27]. From a Bayesian's view, Thompson sampling is proved to be optimal[2] on the stochastic MAB problem. Van Roy et al. gave information theoretic analysis on Thompson sampling for the general cases[16]. A/B testing[39] is another method to deal with exploration-exploitation tradeoff. The multi-armed multi-stage testing borrowed ideas from the MAB algorithms[36]. The *Thompson sampling causal forest*[15] and *random-forest bandit*[18] were non-parametric contextual bandit algorithms, but these works did not provide theoretic regret bound. The paper[20] proposed a non-parametric online bandit algorithm using *k-Nearest-Neighbor*. Our causal-forest based algorithm can improve their bounds in a high-dimensional setting. Our paper differs from the above online bandit algorithms in that we use the logged data to provide a warm-start of these online algorithms, and we derive regret bounds. In addition, we propose one novel contextual bandit algorithm, i.e. ϵ-greedy causal forest, and theoretically analyze its regret.

Several works aimed at using logged data to help online decision making. The historical UCB algorithm[33] was a special case of our algorithmic framework while they ignored users' contexts. As shown by Example 1, ignoring contexts can result in misleading initialization of online algorithms. Authors in [9][19] combined the observational data, experimental data and conterfactual data, to solve the MAB problem with unobserved confounders. They considered a different problem of maximazing the "intent-specific reward", and they did not analyze the regret bound. A recent work studied how to robustly combine supervised learning and online learning[40], and proposed an adaptive weighting algorithm. Supervised learning (curve fitting) discovers correlation instead of causation[30], and the predictions are biased when online data's distribution is different from logged data. Their work focused on correcting the bias of supervised learning via online feedbacks, while our paper uses causal inference methods to unbiasedly initialize online algorithms provided that we observe enough confounders.

8 CONCLUSIONS & FUTURE WORKS

This paper studies the problem of how to use the logged data to make better online decisions. We unify the offline causal inference algorithms and online bandit learning algorithms into a single framework, and consider both context-independent and contextual decisions. We introduce five instances of algorithms that incorporate well-known causal inference methods including matching, weighting, causal forest, and well-known bandit algorithms including UCB and LinUCB. We also propose a new analytical framework. Following this framework, we theoretically analyze the regret bounds for all the algorithmic instances. Experiments on synthetic data validate these theoretical findings. The analysis and experiments show that our algorithms that use both logged data and online feedbacks outperform algorithms that only use one data source. We also show how the quality and quantity of logged data impact the benefits of using logged data.

This paper connects the works of offline causal inference and online bandit learning. Our framework alleviates the cold-start problem of online learning, and implies possibility to design new online algorithms. For example, borrowing the *causal-forest* in causal inference, we propose the *"ϵ-greedy causal-forest"* contextual bandit algorithm and theoretically analyze its regret bound. Experiments show that when linearity property does not hold, our *ϵ-greedy causal-forest* algorithm still significantly outperforms the conventional LinUCB algorithm. Our framework can be applied to all previous applications of offline causal inference and online bandit learning, such as log-assisted online testing system (e.g. A/B tests), recommendation systems[38][27] and online advertising[11].

This work is an initial step towards data driven (online) decisions. We leave the relaxations of our assumptions in future works. First, when the ignorability assumption does not hold, how many logged data samples should we use to make better online decisions? Second, when the stable unit assumption does not hold, there are two important cases: (1) the system has state transitions driven by the actions of the decision maker – this lead to the problem of using causal inference to warm-start reinforcement learning; (2) the system has periodic changes (e.g. changes between day and night) – this requires the online oracle to handle periodic changes.

Appendix

Our appendix consists the proofs of the theorems. For easy references, we copy the theorem statements here.

A THEORETICAL ANALYSIS ON THE REGRET

We first give a general analytical framework for the problem. We then analyze the regret upper bound of the five instantiations of algorithms.

A.1 General analytical framework

Theorem 1 (general upper bound). *Suppose (1) for the online bandit oracle O, there exists a function $g(T)$, such that the regret of only using the online feedbacks $R(T,O)\leq g(T)$ $\forall T$; (2) the offline evaluator $\mathcal{M}$ returns unbiased outcomes, i.e. $\mathbb{E}[\mathcal{M}(\boldsymbol{x},a)]=\mathbb{E}[y|a]$ for the context-independent case, or $\mathbb{E}[\mathcal{M}(\boldsymbol{x},a)]=\mathbb{E}[y|a,\boldsymbol{x}]$ for the contextual case*[1]*. Suppose the offline evaluator returns $\{\tilde{y}_j\}_{j=1}^N$ w.r.t. $\{(\tilde{\boldsymbol{x}}_j,\tilde{a}_j)\}_{j=1}^N$ till time T. Then, for a contextual-independent algorithm $\mathcal{A}$,*

$$R(T,\mathcal{A}) \leq g(T+N) - \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a] - \mathbb{E}[y|a=\tilde{a}_j]\right).$$

For a contextual algorithm $\mathcal{A}_c$, we also have a regret bound :

$$R(T,\mathcal{A}_c) \leq g(T+N) - \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a,\tilde{\boldsymbol{x}}_j] - \mathbb{E}[y|a=\tilde{a}_i,\tilde{\boldsymbol{x}}_i]\right).$$

Proof. The proof follows the idea described in Section 3.2. Online learning oracle is called for $N+T$ times, including N times with synthetic feedbacks and T times with real feedbacks. Denote the total pseudo-regret in these $N+T$ time slots as $R(O,N+T)$. Because the condition (2) ensures that our offline evaluator returns unbiased i.i.d. samples in different time slots, the online bandit oracle cannot distinguish these offline samples from online samples (note that the regret upper bound do not need to rely on higher moments). Then according to the regret bound of the online learning oracle, we have

$$R(O,N+T) \leq g(N+T). \tag{12}$$

Moreover, we could decompose the total regret of the online learning oracle as

$$R(O,N+T) = \sum_{j=1}^{N}(\max_{a\in[K]}\mathbb{E}[y|a] - \mathbb{E}[y|a=\tilde{a}_j]) + R(\mathcal{A},T) \tag{13}$$

On the right hand side, the first term $\sum_{j=1}^{N}(\max_{a\in[K]}\mathbb{E}[y|a] - \mathbb{E}[y|a=\tilde{a}_j])$ is the cummulative regret of the bandit oracle in the offline phase, and the second term $R(\mathcal{A},T)$ is the cumulative regret in the online phase. Combining (12) and (13), we get

$$R(\mathcal{A},T) \leq g(N+T) - \sum_{j=1}^{N}(\mathbb{E}[y|a^*] - \mathbb{E}[y|\tilde{a}_i]),$$

which concludes our proof for the context-independent case. For the contextual case, the proof is similar and we only need to replace $\mathbb{E}[y|a]$ with $\mathbb{E}[y|a,\boldsymbol{x}]$. □

Corollary 1. *Conditions in Theorem 1 holds. Suppose the offline evaluator returns $\{\tilde{y}_j\}_{j=1}^N$ w.r.t. $\{(\tilde{\boldsymbol{x}}_j,\tilde{a}_j)\}_{j=1}^N$ till time T If an online bandit oracle satisfies the "no-regret" property, i.e. $\lim_{T\to+\infty} g(T)/T=0$ (and g is concave), then the difference of regret bounds (before and after using offline data) has the following limit for large T:*

$$\lim_{T\to+\infty} g(T) - R(T,\mathcal{A}) \geq \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a] - \mathbb{E}[y|a=\tilde{a}_j]\right),$$

for context-independent algorithm $\mathcal{A}$. For contextual algorithm $\mathcal{A}_c$,

$$\lim_{T\to+\infty} g(T) - R(T,\mathcal{A}_c) \geq \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a,\tilde{\boldsymbol{x}}_j] - \mathbb{E}[y|a=\tilde{a}_j,\tilde{\boldsymbol{x}}_j]\right).$$

Proof. Based on Theorem 1, we only need to show $\lim_{T\to+\infty} g(N+T) - g(T) = 0$. Before we start our proof, we want to point out that regret bounds of many bandit algorithms have "no-regret" property. For example, the regret bound $g(T)$ for UCB is proportional to $\log(T)$, the regret bound $g(T)$ for EXP3 is proportional to $\sqrt{T}$. These functions w.r.t. T are sub-linear and concave. These functions are concave because as the oracle receives more online feedbacks, it makes better decisions and thus has less regret per time slot. For the concave function, $\frac{g(N+T)-g(T)}{N}$ is decreasing in T. We claim that $\lim_{T\to+\infty}\frac{g(N+T)-g(T)}{N} = 0$. Otherwise, there will be a $l>0$, such that $\frac{g(N+T)-g(T)}{N} \geq l$, for $T\geq T_0$ where T_0 is a constant. It means that gradient of $g(T)$ is larger than l when T is large. Then, $\lim_{T\to+\infty} g(T)/T \geq l$ which contradicts to the "no-regret" property.

Then, $N\times\lim_{T\to+\infty}\frac{g(N+T)-g(T)}{N} = N\times 0 = 0$. Now, we have

$$\begin{aligned}
&\lim_{T\to+\infty} g(T) - R(T,\mathcal{A})\\
=&\lim_{T\to+\infty}(g(T)-g(N+T)) + \lim_{T\to+\infty}(g(N+T)-R(T,\mathcal{A}))\\
\geq&0 + \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a] - \mathbb{E}[y|a=\tilde{a}_j]\right),
\end{aligned}$$

which completes our proof for the context-independent case. For the contextual case, the proof is similar and we only need to replace $\mathbb{E}[y|a]$ with $\mathbb{E}[y|a,\boldsymbol{x}]$. □

Theorem 2 (general lower bound). *Suppose for any online bandit oracle $\widetilde{O}$, there exists a non-decreasing function $h(T)$, such that $R(T,\widetilde{O}) \geq h(T)$ for $\forall T$, where $h(T)$ is the regret lower bound for all possible algorithms. The offline estimator returns unbiased outcomes $\{\tilde{y}_j\}_{j=1}^N$ w.r.t. $\{(\tilde{\boldsymbol{x}}_j,\tilde{a}_j)\}_{j=1}^N$. Then for any contextual-independent algorithm $\widetilde{\mathcal{A}}$ we have:*

$$R(T,\widetilde{\mathcal{A}}) \geq h(T) - \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a] - \mathbb{E}[y|a=\tilde{a}_j]\right).$$

We also have a regret bound for any contextual algorithm $\widetilde{\mathcal{A}}_c$:

$$R(T,\widetilde{\mathcal{A}}_c) \geq h(T) - \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a,\tilde{\boldsymbol{x}}_j] - \mathbb{E}[y|a=\tilde{a}_j,\tilde{\boldsymbol{x}}_j]\right).$$

Proof. After decomposing the total regret to the offline phase and online phase, we have for any bandit oracle $\widetilde{O}$

$$\begin{aligned}
R(T,\mathcal{A}) &= R(T+N,\widetilde{O}) - \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a] - \mathbb{E}[y|a=\tilde{a}_j]\right)\\
&\geq h(T+N) - \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a] - \mathbb{E}[y|a=\tilde{a}_j]\right).
\end{aligned} \tag{14}$$

Next, for a non-decreasing function $h(\cdot)$ we have

$$h(T+N) \geq h(T). \tag{15}$$

Combining (14) and (15), we have

$$R(T,\mathcal{A}) \geq h(T) - \sum_{j=1}^{N}\left(\max_{a\in[K]}\mathbb{E}[y|a] - \mathbb{E}[y|a=\tilde{a}_j]\right)$$

which concludes our proof for the unbiased estimators. For the contextual case, the proof is similar and we only need to replace $\mathbb{E}[y|a]$ with $\mathbb{E}[y|a,\boldsymbol{x}]$. □

[1] Some bandit algorithm (e.g. UCB) will assume that the reward is bounded or follow certain distribution like sub-Gaussian. These assumptions also hold for the offline samples, since we consider that the offline samples can represent the online environment.

A.2 Regret bounds for the context-independent algorithms

Theorem 3 (Exact matching + UCB). *Assumptions 1, 2, 3 hold. Suppose there are C possible categories of users' features $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^C$. Denote $\widehat{\mathbb{P}}[\boldsymbol{x}^c]$ as the fraction of online users whose context is $\boldsymbol{x}^c$, whose expectation is $\mathbb{P}[\boldsymbol{x}^c] \triangleq \mathbb{E}[\widehat{\mathbb{P}}[\boldsymbol{x}^c]]$. Denote $a^* \triangleq \arg\max_{\tilde{a}\in[K]} \mathbb{E}[y|\tilde{a}]$, $\Delta_a \triangleq \mathbb{E}[y|a^*] - \mathbb{E}[y|a]$. Let $N(\boldsymbol{x}^c, a) \triangleq \sum_{i\in[-I]} \mathbb{1}_{\{\boldsymbol{x}_i=\boldsymbol{x}^c, a_i=a\}}$ be the number of samples with context $\boldsymbol{x}^c$ and action a. Then,*

$$R(T, \mathcal{A}_1) \leq \sum_{a\neq a^*} \left(1 + \frac{\pi^2}{3} + \sum_{c\in[C]} \max\left\{0, 8\frac{\ln(T+A)}{\Delta_a^2}\widehat{\mathbb{P}}[\boldsymbol{x}^c] - \min_{\tilde{c}\in[C]} \frac{N(\boldsymbol{x}^{\tilde{c}}, a)\mathbb{P}[\boldsymbol{x}^c]}{\mathbb{P}[\boldsymbol{x}^{\tilde{c}}]}\right\}\right)\Delta_a,$$

where the constant

$$A = N - \sum_{a\neq a^*} \sum_{c\in[C]} \max\left\{0, N(\boldsymbol{x}^c, a) - (8\frac{\ln(T+N)}{\Delta_a^2} + 1 + \frac{\pi^2}{3})\mathbb{P}[\boldsymbol{x}^c]\right\}.$$

Proof of Theorem 3. The idea of the proof is similar to that for the general bounds and Appendix A.1. According to Assumption 2 (ignorability), the exact-matching offline evaluator returns unbiased outcomes. Since all the decisions are made by the online learning oracle, we can apply the regret bound of the UCB algorithm, and minus the regrets of *virtual* plays for the samples returned by the exact matching evaluator. Suppose λ_a is the expected number of rounds that the a_{th} arm is pulled by the online learning oracle which is a random number. Let M_a be the number of times that the offline evaluator returns the a_{th} arm. Then, the expected regret

$$R(\mathcal{A}_1, T) = \sum_{a\in[K]} \mathbb{E}[(\lambda_a - M_a)]\Delta_a. \tag{16}$$

Now, we count the number of times M_a that an action a is matched by the exact matching offline evaluator. Denote $M(\boldsymbol{x}^c, a)$ as the number of times $(\boldsymbol{x}^c, a)$ is matched by the offline evaluator, hence $\sum_{c\in[C]} M(\boldsymbol{x}^c, a) = M_a$. We consider the following two cases: (1) the matching process does not terminate at T. In this case the expected number $\mathbb{E}[M_a] = \lambda_a\widehat{\mathbb{P}}$, because the context and action are generated independently. (2) the matching process terminates before T. In this case, we run out of the samples with $(\boldsymbol{x}^{\tilde{c}}, a)$. Suppose the unmatched context is $\boldsymbol{x}^{\tilde{c}}$, then the expected number of matched sample for some other context $\boldsymbol{x}^{\tilde{c}}$ is $N(\boldsymbol{x}^{\tilde{c}}, a)\frac{\mathbb{P}[\boldsymbol{x}^c]}{\mathbb{P}[\boldsymbol{x}^{\tilde{c}}]}$. Consider the worst case over all contexts, then $M(\boldsymbol{x}^c, a) \geq \min_{\tilde{c}\in[C]} N(\boldsymbol{x}^{\tilde{c}}, a)\frac{\mathbb{P}[\boldsymbol{x}^c]}{\mathbb{P}[\boldsymbol{x}^{\tilde{c}}]}$. Note that when $\tilde{c} = c$, $\frac{N(\boldsymbol{x}^{\tilde{c}}, a)\mathbb{P}[\boldsymbol{x}^c]}{\mathbb{P}[\boldsymbol{x}^{\tilde{c}}]} = N(\boldsymbol{x}^{\tilde{c}})$. Combining the counts of M_a in these two cases, we have

$$\mathbb{E}[M_a] \geq \sum_{c\in[C]} \min\left\{\min_{\tilde{c}\in[C]} \frac{N(\boldsymbol{x}^{\tilde{c}}, a)\mathbb{P}[\boldsymbol{x}^c]}{\mathbb{P}[\boldsymbol{x}^{\tilde{c}}]}, \lambda_a\widehat{\mathbb{P}}[\boldsymbol{x}^c]\right\}. \tag{17}$$

Combine (16) and (17), and we decompose $\lambda_a = \sum_{c\in[C]} \widehat{\mathbb{P}}[\boldsymbol{x}^c]$, then

$$R(\mathcal{A}_1, T) \leq \sum_{a\in[K]} \left(\sum_{c\in[C]} \mathbb{E}\left[\max\{\lambda_a\widehat{\mathbb{P}}[\boldsymbol{x}^c] - \min_{\tilde{c}\in[C]} \frac{N(\boldsymbol{x}^{\tilde{c}}, a)\mathbb{P}[\boldsymbol{x}^c]}{\mathbb{P}[\boldsymbol{x}^{\tilde{c}}]}, 0\}\right]\right)\Delta_a$$

We have the following equality:

$$\begin{aligned}
&\max\{\lambda_a\widehat{\mathbb{P}}[\boldsymbol{x}^c] - \min_{\tilde{c}\in[C]} \frac{N(\boldsymbol{x}^{\tilde{c}}, a)\mathbb{P}[\boldsymbol{x}^c]}{\mathbb{P}[\boldsymbol{x}^{\tilde{c}}]}, 0\} \\
&= \max\{l_a\widehat{\mathbb{P}}[\boldsymbol{x}^c] + (\lambda_a - l_a)\widehat{\mathbb{P}}[\boldsymbol{x}^c] - \min_{\tilde{c}\in[C]} \frac{N(\boldsymbol{x}^{\tilde{c}}, a)\mathbb{P}[\boldsymbol{x}^c]}{\mathbb{P}[\boldsymbol{x}^{\tilde{c}}]}, 0\} \\
&= \max\{l_a\widehat{\mathbb{P}}[\boldsymbol{x}^c] - \min_{\tilde{c}\in[C]} \frac{N(\boldsymbol{x}^{\tilde{c}}, a)\mathbb{P}[\boldsymbol{x}^c]}{\mathbb{P}[\boldsymbol{x}^{\tilde{c}}]}, 0\} + (\lambda_a - l_a)\widehat{\mathbb{P}}[\boldsymbol{x}^c]
\end{aligned}$$

We define

$$l_a = \lceil(8\ln(T + \mathbb{E}[\sum_{a\in[K]} M_a]))/\Delta_a^2\rceil. \tag{18}$$

Then, $l_a \geq \mathbb{E}[\lceil\mathbb{E}[8\ln(T + \sum_{a\in[K]} M_a)]\rceil]$ because $\ln(\cdot)$ is a convex function (according to Jensen's inequality). According Assumption 1 and 3 (stable unit), we can apply the results in paper of Auer et al.[6] and $\mathbb{E}[\lambda_a - l_a] \leq 1 + \frac{\pi^2}{3}$ for some sub-optimal action $a \neq a^*$. Therefore, we have

$$R(\mathcal{A}_1, T) \leq \sum_{a\in[K]} \left((1 + \frac{\pi^2}{3}) + \sum_{c\in[C]} \max\{l_a\widehat{\mathbb{P}}[\boldsymbol{x}^c] - \min_{\tilde{c}\in[C]} \frac{N(\boldsymbol{x}^{\tilde{c}}, a)\mathbb{P}[\boldsymbol{x}^c]}{\mathbb{P}[\boldsymbol{x}^{\tilde{c}}]}, 0\}\right)\Delta_a \tag{19}$$

To get an upper bound for l_a, we now give an upper bound for the expected number of samples that are matched, i.e. $\mathbb{E}[\sum_{a\in[K]} M_a]$. Recall that we denote the number of matched samples with context $\boldsymbol{x}^c$ and arm j as $M(\boldsymbol{x}^c, a)$. Then, $\mathbb{E}[M(\boldsymbol{x}^c, a)] \leq N(\boldsymbol{x}^c, a)$ because it cannot exceed the number of data samples. Also, $\mathbb{E}[M(\boldsymbol{x}^c, a)] \leq \mathbb{E}[\lambda_a]\mathbb{P}[\boldsymbol{x}^c]$ because the expected number of matched samples cannot exceed the expected number of times the action is selected. Therefore, $\mathbb{E}[M(\boldsymbol{x}^c, a)] \leq \max\{N(\boldsymbol{x}^c, a), \lambda_a\mathbb{P}[\boldsymbol{x}^c]\}$. Then, we have

$$\begin{aligned}
&\mathbb{E}[\sum_{a\in[K]} M_a] \leq \sum_{c\in[C]} \sum_{a\in[K]} \min\{N(\boldsymbol{x}^c, a), \lambda_a\mathbb{P}[\boldsymbol{x}^c]\} \\
&= -\sum_{c\in[C]} \sum_{a\in[K]} \max\{-N(\boldsymbol{x}^c, a), -\lambda_a\mathbb{P}[\boldsymbol{x}^c]\} \\
&= \sum_{c=1}^{C}\sum_{a=1}^{K} N(\boldsymbol{x}^c, a) - \sum_{c=1}^{C}\sum_{a=1}^{K} \max\left\{N(\boldsymbol{x}^c, a) - N(\boldsymbol{x}^c, a), N(\boldsymbol{x}^c, a) - \mathbb{E}[\lambda_a]\mathbb{P}[\boldsymbol{x}^c]\right\} \\
&= N - \sum_{c\in[C]} \sum_{a\in[K]} \max\{0, N(\boldsymbol{x}^c, a) - \mathbb{E}[\lambda_a]\mathbb{P}[\boldsymbol{x}^c]\} \\
&\leq N - \sum_{c\in[C]} \sum_{a\in[K]} \max\{0, N(\boldsymbol{x}^c, a) - (8\frac{\ln(T+N)}{\Delta_a^2} + 1 + \frac{\pi^2}{3})\mathbb{P}[\boldsymbol{x}^c]\}.
\end{aligned} \tag{20}$$

The last equation is because $\mathbb{E}[\lambda_a] \leq 8\frac{\ln(T+N)}{\Delta_a^2} + 1 + \frac{\pi^2}{3}$ according to paper[6].

Plug-in (18) and (20) to (19), then we have the upper bound claimed by our Theorem. □

Theorem 4 (Propensity score matching + UCB). *Assumptions 1, 2, 3 hold. We consider that the propensity scores are in a finite set $p_i \in \boldsymbol{Q} \triangleq \{q_1, \ldots, q_Q\} \subseteq [0, 1]$ for $\forall i \in [-I]$. Let $N(q, a)$ be the number of data items whose propensity score $p_i = q$ and action $a_i = a$. Denote $\widehat{\mathbb{P}}[q_c]$ as the fraction of online users whose propensity score is q_c. Then,*

$$R(T,\mathcal{A}_2)\leq\sum_{a\neq a^*}\left(1+\frac{\pi^2}{3}+\sum_{c\in[Q]}\max\left\{0,8\frac{\ln(T+A)}{\Delta_a^2}\widehat{\mathbb{P}}[q_c]-\min_{\tilde{c}\in[Q]}\frac{N(q_{\tilde{c}},a)\mathbb{P}[q_c]}{\mathbb{P}[q_{\tilde{c}}]}\right\}\right)\Delta_a,$$

where $\mathbb{P}[q_c]\triangleq\mathbb{E}[\widehat{\mathbb{P}}[q_c]]$ *is the probability for propensity score* q_c*, and*

$$A=N-\sum_{a\neq a^*}\sum_{c\in[Q]}\max\left\{0,\ \min_{\tilde{c}\in[Q]}\frac{N(q_{\tilde{c}},a)\mathbb{P}[q_c]}{\mathbb{P}[q_{\tilde{c}}]}-(8\frac{\ln(T+N)}{\Delta_a^2}+1+\frac{\pi^2}{3})\mathbb{P}[q_c]\right\}.$$

Proof. The proof is similar from the proof of exact matching. The only difference is that for propensity score matching, the features to be matched contain only the propensity score.

First, we will show that by matching the propensity score, the expected reward in each round for each arm is not changed.

The expected reward when we choose action a is

$$\mathbb{E}[y|a]=\sum_{\boldsymbol{x}\in\mathcal{X}}\mathbb{P}[\boldsymbol{x}]\mathbb{E}[y|a,\boldsymbol{x}],$$

where $\mathbb{E}[y|a,\boldsymbol{x}]$ is the expected reward when the context is $\boldsymbol{x}$ and the action is a. We then consider the expected reward when we use the propensity score matching strategy. Let us denote the propensity score of choosing an action $\tilde{a}$ under context $\tilde{\boldsymbol{x}}$ as

$$p(\tilde{\boldsymbol{x}},\tilde{a})=\mathbb{P}[a=\tilde{a}|\boldsymbol{x}=\tilde{\boldsymbol{x}}].$$

The expected reward of choosing an action $\tilde{a}$ is

$$\begin{aligned}&\sum_{\boldsymbol{x}\in\mathcal{X}}\mathbb{P}[\boldsymbol{x}]\mathbb{E}[y|p{=}p(\boldsymbol{x},\tilde{a}),a{=}\tilde{a}]\\&=\sum_{c\in[Q]}\sum_{\boldsymbol{x}\in\mathcal{X}}\mathbb{P}[\boldsymbol{x}]\mathbb{1}_{\{p(\boldsymbol{x},i)=p_c\}}\mathbb{E}[y|p{=}p_c,a{=}\tilde{a}].\end{aligned}$$

and we have

$$\begin{aligned}\mathbb{E}[y|p{=}p_c,a{=}\tilde{a}]&=\frac{\sum_{\boldsymbol{x}\in\mathcal{X}}\mathbb{E}[y|\boldsymbol{x},\tilde{a}]\times\mathbb{P}[\boldsymbol{x}]\mathbb{1}_{\{p(\boldsymbol{x},\tilde{a})=p_c\}}p_c}{\sum_{\boldsymbol{x}\in\mathcal{X}}\mathbb{P}[\boldsymbol{x}]\times\mathbb{1}_{\{p(\boldsymbol{x},\tilde{a})=p_c\}}p_c}\\&=\frac{\sum_{\boldsymbol{x}\in\mathcal{X}}\mathbb{E}[y|\boldsymbol{x},\tilde{a}]\times\mathbb{P}[\boldsymbol{x}]\mathbb{1}_{\{p(\boldsymbol{x},\tilde{a})=p_c\}}}{\sum_{\boldsymbol{x}\in\mathcal{X}}\mathbb{P}[\boldsymbol{x}]\times\mathbb{1}_{\{p(\boldsymbol{x},\tilde{a})=p_c\}}}\end{aligned}$$

Therefore, we have

$$\begin{aligned}&\sum_{\boldsymbol{x}\in\mathcal{X}}\mathbb{P}[\boldsymbol{x}]\mathbb{E}[y|p{=}p(\boldsymbol{x},\tilde{a}),a{=}\tilde{a}]\\&=\sum_{c\in[Q]}\sum_{\boldsymbol{x}\in\mathcal{X}}\mathbb{P}[\boldsymbol{x}]\mathbb{1}_{\{p(\boldsymbol{x},\tilde{a})=p_c\}}\frac{\sum_{\boldsymbol{x}\in\mathcal{X}}p(\boldsymbol{x},\tilde{a})\mathbb{P}[\boldsymbol{x}]\mathbb{1}_{\{p(\boldsymbol{x},\tilde{a})=p_c\}}}{\sum_{\boldsymbol{x}\in\mathcal{X}}\mathbb{P}[\boldsymbol{x}]\mathbb{1}_{\{p(\boldsymbol{x},\tilde{a})=p_c\}}}\\&=\sum_{c\in[Q]}\sum_{\boldsymbol{x}\in\mathcal{X}}p(\boldsymbol{x},\tilde{a})\mathbb{P}[\boldsymbol{x}]\mathbb{1}_{\{p(\boldsymbol{x},\tilde{a})=p_c\}}=\sum_{\boldsymbol{x}\in\mathcal{X}}p(\boldsymbol{x},\tilde{a})\mathbb{P}[\boldsymbol{x}]=\mathbb{E}[y|\tilde{a}].\end{aligned}$$

The last but one equation is from our assumption that all the propensity scores are belong to a finite set $\{p_1,\ldots,p_Q\}$, and thus $\sum_{c\in[Q]}\mathbb{1}_{\{p(\boldsymbol{x},\tilde{a})=p_c\}}=1$ (namely, the propensity score belongs to some value in the set).

Hence, our propensity score matching method yields an unbiased estimation of $\mathbb{E}[y|\tilde{a}]$ for any action $\tilde{a}$.

With such unbiasedness property, the remaining is the same as the last theorem, except that the contexts $\boldsymbol{x}$ are changed to the propensity score p. □

The propensity score matching algorithm can improve the probability for the historical data points to be matched, compared to the regret bound for the exact matching algorithm.

Theorem 5 (Inverse propensity score weighting + UCB). *Assumptions 1, 2, 3 hold. Suppose the online reward is bounded* $y_t\in[0,1]$ $\forall t\in[T]$ *and the propensity score is bounded* $p_i\geq\bar{s}>0$ *for* $\forall i\in[I]$*. Then*

$$R(T,\mathcal{A}_3)\leq\sum_{a\neq a^*}\Delta_a\left(\max\left\{0,8\frac{\ln(T+\sum_{a=1}^K\lceil N_a\rceil)}{\Delta_a^2}-\lfloor N_a\rfloor\right\}+(1+\frac{\pi^2}{3})\right)$$

where we recall $N_a=\left(\sum_{i\in[-I]}\frac{1}{p_i}\mathbb{1}_{\{a_i=a\}}\right)^2/\sum_{i\in[-I]}\left(\frac{1}{p_i}\mathbb{1}_{\{a_i=a\}}\right)^2$.

Proof. The proof follows the same idea as previous ones. We will first show that the estimation relying on the offline data is unbiased. Second, we use a weighted Chernoff bound to show the effective number of logged samples in terms of the confidence bound.

Many previous works have shown the inverse propensity weighting method provides an unbiased estimator[35]. In fact, for $\tilde{a}\in[K]$

$$\begin{aligned}\mathbb{E}[\bar{y}_{\tilde{a}}]&=\frac{\mathbb{E}[\sum_{i\in[-I]}\mathbb{E}[y|\boldsymbol{x}_i,\tilde{a})]\mathbb{E}[\mathbb{1}_{\{a_i=\tilde{a}\}}]/p(\boldsymbol{x}_i,\tilde{a})]}{\sum_{i\in[-I]}\mathbb{E}[\mathbb{1}_{\{a_i=\tilde{a}\}}]/p(\boldsymbol{x}_i,\tilde{a})]}\\&=\frac{\mathbb{E}[\sum_{i\in[-I]}\mathbb{E}[y|\boldsymbol{x}_i,\tilde{a}]]}{I}\\&=\sum_{\boldsymbol{x}\in\mathcal{X}}\mathbb{P}[\boldsymbol{x}]\mathbb{E}[y|\boldsymbol{x},\tilde{a}]=\mathbb{E}[\bar{y}_{\tilde{a}}].\end{aligned}$$

The second equation holds because the probability that we observe the action $\tilde{a}$ is $\mathbb{E}[\mathbb{1}_{\{a_i=\tilde{a}\}}]$ which is the propensity score $p(\boldsymbol{x}_i,\tilde{a})$. The last equation is because the expectation for data item i is taken over the contexts $\boldsymbol{x}$.

According to Chernoff-Hoeffding bound [22], we have

Lemma 9. *If* $X_1,X_2,\ldots,X_n$ *are independent random variables and* $A_i\leq X_i\leq B_i(i=1,2,\ldots,n)$*, we have the following bounds for the sum* $X=\sum_{i=1}^n X_i$*:*

$$\mathbb{P}[X\leq\mathbb{E}[X]-\delta]\leq e^{-\frac{2\delta^2}{\sum_{i=1}^n(B_i-A_i)^2}}.$$

$$\mathbb{P}[X\geq\mathbb{E}[X]+\delta]\leq e^{-\frac{2\delta^2}{\sum_{i=1}^n(B_i-A_i)^2}}.$$

In our case to estimate the outcome for an action a, we have $X_i=y_i\frac{\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)}{\sum_{i\in[-I]}\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)}$, and $X=\sum_{i\in[-I]}X_i=\bar{y}_a$. Hence the constants $A_i=0$, $B_i=\frac{\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)}{\sum_{i\in[-I]}\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)}$. Therefore, we have

$$\begin{aligned}&\mathbb{P}[|\bar{y}_a-\mathbb{E}[y|a]|\geq\delta]\\&\leq 2e^{-\frac{2\delta^2}{\sum_{i\in[-I]}\left(\frac{\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)}{\sum_{i\in[-I]}\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)}\right)^2}}\\&=2e^{-\frac{2\delta^2}{\frac{\sum_{i\in[-I]}\left(\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)\right)^2}{\left(\sum_{i\in[-I]}\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)\right)^2}}}\\&=2e^{-2\delta^2\frac{\left(\sum_{i\in[-I]}\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)\right)^2}{\sum_{i\in[-I]}\left(\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)\right)^2}}\end{aligned}$$

We compare it with the Chernoff-Hoeffding bound used in the UCB algorithm[6]. When we have n_a online samples of arm a,

$$\mathbb{P}[|\bar{y}_a-\mathbb{E}[y|a]|\geq\delta]\leq 2e^{-2n_a\delta^2}.$$

By this comparison, we let $n=\widehat{N}_a$ and we will get the same bound.

Now, we show that by using these $\lfloor\widehat{N}_a\rfloor$ samples from logged data, the online bandit UCB oracle will always have a tighter bound than that for $\lfloor\widehat{N}_a\rfloor$ i.i.d. samples from the online environment.

In the online phase, let the number of times to play the action a to be T_a. For the offline samples, let $X_i=y_i\frac{\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)}{\sum_{i\in[-I]}\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)}\frac{\widehat{N}_a}{\widehat{N}_a+T_a}$.

For the online samples, let $X^t = y_t \frac{1}{\widehat{N}_a + T_a}$. Let us consider the sequence $\{X_1, \ldots, X_I, X^1, \ldots, X^{T_a}\}$. Now, $X = \sum_{i\in[-I]} X_i + \sum_{t\in[T_a]} X^t$. Then, we have $\mathbb{E}[X] = \mathbb{E}[y|a]$, and $0 \le X_i \le \frac{\widehat{N}_a}{\widehat{N}_a+T_a} B_i (i \in [-I])$, $0 \le X^t \le \frac{1}{\widehat{N}_a+T_a}$. In addition, we have

$$\left(\frac{\widehat{N}_a}{\widehat{N}_a+T_a}\right)^2 \frac{\sum_{i\in[-I]} \left(\mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)\right)^2}{\sum_{i\in[-I]} \mathbb{1}_{\{a_i=a\}}/p(\boldsymbol{x}_i,a_i)} + \sum_{t\in[T_a]} \left(\frac{1}{\widehat{N}_a+T_a}\right)^2$$

$$= \left(\frac{\widehat{N}_a}{\widehat{N}_a+T_a}\right)^2 \left(\frac{1}{\widehat{N}_a}\right) + \frac{T_a}{(\widehat{N}_a+T_a)^2} = \frac{1}{\widehat{N}_a+T_a}.$$

Therefore,

$$\mathbb{P}[\bar{y}_a \le \mathbb{E}[y|a] - \delta] \le e^{-2\delta^2(\widehat{N}_a+T_a)},$$
$$\mathbb{P}[\bar{y}_a \ge \mathbb{E}[y|a] + \delta] \le e^{-2\delta^2(\widehat{N}_a+T_a)}.$$

In other words, when we have T_a online samples of an action a, the confidence interval is as if we have $T_a + \widehat{N}_a$ total samples for the bandit oracle. Then, the regret bound reduces to the case where we have $\widehat{N}_a$ offline samples for arm a that do not have contexts. □

A.3 Regret bounds for contextual algorithms

For the contextual case, we first analyze the linear-model-based algorithm $\mathcal{A}_4$, and then analyze the forest-based algorithm $\mathcal{A}_5$.

$\mathcal{A}_3$: linear regression + LinUCB. For this linear-model-based algorithm, we first consider the *problem-dependent* regret bound (which is in our main paper) and then consider a *problem-independent* regret bound (which is NOT in our main paper).

Theorem 7 (Linear regression + LinUCB, problem dependent). *Assumptions 1, 2, 3 hold. In addition, the rewards satisfy the linear model in (2). Suppose offline evaluator returns a sequence $\{y_i\}_{i=1}^N$ w.r.t. $\{(\boldsymbol{x}_i, a_i)\}_{i=1}^N$. Let $V_N \triangleq \sum_{i\in[N]} \boldsymbol{x}_i \boldsymbol{x}_i'$, $L \triangleq \max_{t\le T}\{||\boldsymbol{x}_t||_2\}$. Suppose the outcome is bounded in $[0,1]$. Then*

$$R(T, \mathcal{A}_4) \le \frac{8d(1+2\ln(T))}{\Delta_{\min}} d\log(1+\kappa) + 1,$$

where $\kappa = TL^2/\lambda_{\min}(V_N)$. In particular, when the smallest eigenvalue $\lambda_{\min}(V_N) \ge (1/2 + \ln(T))TL^2$, the regret is bounded by $16d^2/\Delta_{\min}+1$.

Proof. We will first show a high-probability bound, i.e. with probability at least $1-\delta$, the cummulative regret has the bound

$$R(T, \mathcal{A}_4) \le \frac{4\beta_{N+T}(\delta)}{\Delta_{\min}} d\log(1+\kappa)$$

when the parameters $\{\beta_t\}_{t=1}^T$ ensure the confidence bound in each time slot.

Recall that the contexts of samples returned by the offline evaluator are $\boldsymbol{x}_{-1}, \boldsymbol{x}_{-2}, \ldots, \boldsymbol{x}_{-N}$. We denote $r_t \triangleq \max_{a\in[K]} \mathbb{E}[y_t|\boldsymbol{x}_t, a] - \mathbb{E}[y_t|\boldsymbol{x}_t, a_t]$ as the pseudo-regret in time slot t. Recall that $\beta_t(\delta)$ is the parameter β_t in the t^{th} time slot, and the δ is to emphasize that it is a function of δ. From the proof for the problem-independent bound in paper[1], we know $\sum_{t=1}^T r_t \le \frac{4\beta_{N+T}(\delta)}{\Delta_{\min}} \log \frac{\det V_T}{\det V_N}$. The following is to bound $\log \frac{\det V_{N+T}}{\det V_N}$. We have the following lemma.

Lemma 10. *Let $\kappa = \frac{TL^2}{\lambda_{\min}(V_N)}$, then $(1+\kappa)V_N \succcurlyeq V_{T+N}$.*

Proof. We first consider the case where all the data samples are returned before the first online phase start. Denote the V matrix in the online time slot t after using the logged data as V_{N+t}. Note that $V_{T+N} = V_N + \sum_{t=1}^T \boldsymbol{x}_t \boldsymbol{x}_t'$. Thus the above lemma is equivalent to $\sum_{t=1}^T \boldsymbol{x}_{N+t} \boldsymbol{x}_{N+t}' \preccurlyeq \kappa V_N$. Here, we use $\boldsymbol{x}'$ to denote the transpose of $\boldsymbol{x}$ (to avoid using "$\boldsymbol{x}^T$" with the confusing T). The positive semi-definiteness means that for any $\boldsymbol{x}$ where $||\boldsymbol{x}||_2$=1, we want to have

$$\boldsymbol{x}' \left(\sum_{t=1}^T \boldsymbol{x}_t \boldsymbol{x}_t'\right) \boldsymbol{x} \le \kappa \boldsymbol{x}' V_N \boldsymbol{x}. \tag{21}$$

In fact $\boldsymbol{x}' \left(\sum_{t=1}^T \boldsymbol{x}_t \boldsymbol{x}_t'\right) \boldsymbol{x} \le TL^2$, because L is the maximum 2-norm of $\boldsymbol{x}_t$. In addition, $\boldsymbol{x}' V_N \boldsymbol{x} \ge \lambda_{\min}(V_N)$. Hence, we always have (21) for $\forall \boldsymbol{x}$. Hence we proved the above lemma. □

We have $\det A \le \det B$ if $A \preccurlyeq B$. Hence,

$$\det V_{T+N} \le \det(1+\kappa)V_N = (1+\kappa)^d \det V_N.$$

Then, $\log \frac{\det V_{N+T}}{\det V_N} \le d\log(1+\kappa)$, which leads to our Theorem.

Now, we set $\beta_t(\delta) = 2d(1 + 2\ln(1/\delta))$, and the parameter is in the confidence ball with probability at least $1-\delta$. Moreover, we set $\delta = 1/T$. Then, the regret in each time slot can be divided into two parts: (1) the δ probability part (summing up to at most 1, because the outcome is bounded); and (2) the $1-\delta$ probability part (summing up to at most $\frac{8d(1+2\ln(T))}{\Delta_{\min}} d\log(1+\kappa)$). Therefore, the expected cumulative reward has an upper bound $\frac{8d(1+2\ln(T))}{\Delta_{\min}} d\log(1+\kappa)+1$. □

Causal forest. Now, we analyze the causal forest algorithm. The analysis is from the confidence bound from Wager and Athey's paper[37][4] on the causal forest.

We first consider the asymptotic regret bound for the online version of the causal forest algorithm. In this case, we do not know the exact structure of the tree and the estimator is potentially biased. In this case, analyzing the regret needs to deal with the bias-variance tradeoff. Recall that in the causal forest algorithm, we use the ϵ-decreasing exploration strategy. Note that our following analysis is for the *ϵ-greedy-causal-forest* online oracle which only uses the online feedbacks (not the algorithm that uses both data sources).

Theorem 8 (ϵ-greedy-causal-forest, asymptotic). *Assumptions 1, 2, 3 hold. The logged data satisfies the conditions of Theorem 1 in paper[37]. In particular, the tree-predictors are α-regular, i.e. "each split leaves at least a fraction $\alpha \le 0.5$ of the available training examples on each side." When the exploration probability $\epsilon_t = t^{-1/2(1-\beta)}$, with no logged data, the asymptotic regret (for any small $\xi > 0$)*

$$\lim_{T\to+\infty} R(T, \mathcal{A}_5)/T^{(2+\xi)/(3-\beta)} < +\infty$$

where $\beta = 1 - \frac{2A}{2+3A}$ and $A = \frac{\pi}{d} \frac{\log((1-\alpha)^{-1})}{\log(\alpha^{-1})}$. Hence $\lim_{T\to+\infty} R(T, \mathcal{A}_5)/T = 0$.

Proof. As a first step, we consider a fixed $\epsilon = T^{\frac{1}{2}(\beta-1)}$ in all rounds, which depends on the number of time slots T. This corresponds to the setting where the decision maker knows the number of rounds in the beginning. For this case, we have the following lemma

Lemma 11. *Consider the same conditions as that in Theorem 8. But we set $\epsilon_t = \epsilon = T^{-\frac{A}{2+3A}}$ as a constant in all the rounds. Then for $\beta = 1 - \frac{2A}{2+3A}$, we have the same asymptotic regret (for any small $\xi > 0$)*

$$\lim_{T\to+\infty} R(T, \mathcal{A}_5)/T^{(2+\xi)/(3-\beta)} < +\infty.$$

Proof. The following proofs extend the results of paper[37] to an online setting. The general idea is to use the "asymptotic unbiased"

and "asymptotic Gaussian" property of causal forest provided that the exploration probability ϵ is high enough. The parameter β balances the bias and variance, and the parameter ϵ balances the exploration probability and the convergence for bias and variance. The decision maker's regret depends on the bias, variance and exploration probability. By setting β and ϵ to appropriate values, we can achieve optimal rate of regret increment.

The results in paper[37] requires $\beta \in \left(1-(1+\frac{d}{\pi}\frac{\log(\alpha^{-1})}{\log((1-\alpha)^{-1})})^{-1}, 1\right)$. Namely, $\beta \in (\frac{1}{A+1}, 1)$. In fact, our choice of $\beta = 1 - \frac{2A}{2+3A} > \frac{1}{A+1}$ satisfy this condition (one can see $1 - \frac{2A}{2+3A} > 1 - \frac{A}{A+1}$ when $A > 0$).

Let $\mu(\boldsymbol{x}, a) \triangleq \mathbb{E}[y|\boldsymbol{x}, a]$ denote the expected outcome of action a under the context $\boldsymbol{x}$, and let $\hat{\mu}(\boldsymbol{x}, a)$ denote its estimated value by causal forest. We use $\tau(\boldsymbol{x})$ to denote the expected *"treatment effect"* $\mu(\boldsymbol{x}, 1) - \mu(\boldsymbol{x}, 0)$. We use $\hat{\tau}(\boldsymbol{x})$ to denotes the estimated *treatment effect* by the causal forest. To help the readers find corresponding theorems in paper[37], we consider the treatment effect $\tau(\boldsymbol{x})$ to present the analysis on bias and variance. One can apply the results on $\tau(\boldsymbol{x})$ to $\mu(\boldsymbol{x}, a)$, since we can set $y|_{(\boldsymbol{x},0)} = 0$ and $\tau(\boldsymbol{x})$ and in this case $\mu(\boldsymbol{x}, 1) = \tau(\boldsymbol{x})$[2].

First, we consider the bias of the causal forest estimator. According to the proof of Theorem 11 in paper[37], we have for some constant M, the bias is bounded by

$$|\mathbb{E}[\hat{\tau}(\boldsymbol{x})] - \tau(\boldsymbol{x})| \lesssim 2Md\left(\frac{\varepsilon s}{2k-1}\right)^{-\frac{1}{2}\frac{\log\left((1-\alpha)^{-1}\right)}{\log(\alpha^{-1})}\frac{\pi}{d}}$$

where $s \triangleq n^{\beta}$, and $f(s) \lesssim g(s)$ if $\lim_{s\to+\infty}\frac{f(s)}{g(s)}$. It means that there exists an integer $N_1 > 0$ and a constant $C_1 > 0$, such that for any $n \geq N_1$ (and s is a function of n), we have

$$|\mathbb{E}[\hat{\tau}(\boldsymbol{x})] - \tau(\boldsymbol{x})| \leq C_1 2Md\left(\frac{\varepsilon s}{2k-1}\right)^{-\frac{1}{2}\frac{\log\left((1-\alpha)^{-1}\right)}{\log(\alpha^{-1})}\frac{\pi}{d}}. \quad (22)$$

Second, we analyze the variance of the estimator. We have the an asymptoticly normal results by Theorem 11 (and Theorem 8) that $\frac{\mathbb{E}[\hat{\tau}(\boldsymbol{x})]-\hat{\tau}(\boldsymbol{x})}{\sigma_n(\boldsymbol{x})} \implies \mathcal{N}(0,1)$. In particular, according to the proof of Theorem 8 in paper[37], $\sigma_n(\boldsymbol{x})^2 \leq \frac{s}{n}Var[T]$ where $Var[T]$ is finite. Therefore, we have a sharper confidence bound based on the Gaussian distribution. We have the following lemma.

Lemma 12. *There exists a $N_2 > 0$, such that for any $n > N_2$, we have for any $\delta > 0$*

$$\mathbb{P}[|\hat{\tau}(\boldsymbol{x}) - \mathbb{E}[\hat{\tau}(\boldsymbol{x})]| \leq \sigma_n(\boldsymbol{x})\delta] \geq 1 - \frac{1}{2}e^{-\delta^2/2}. \quad (23)$$

In addition, $|\hat{\tau}(\boldsymbol{x})-\tau(\boldsymbol{x})| \leq |\hat{\tau}(\boldsymbol{x})-\mathbb{E}[\hat{\tau}(\boldsymbol{x})]|+|\mathbb{E}[\hat{\tau}(\boldsymbol{x})]-\tau(\boldsymbol{x})|$. Now we combine (22) and (23). When $n > \max\{N_1, N_2\}$, with probability at least $1 - \frac{1}{2}e^{-\delta^2/2}$, we have

$$|\hat{\tau}(\boldsymbol{x}) - \tau(\boldsymbol{x})| \leq \sigma_n(\boldsymbol{x})\delta + 2Md\left(\frac{\varepsilon s}{2k-1}\right)^{-\frac{1}{2}\frac{\log\left((1-\alpha)^{-1}\right)}{\log(\alpha^{-1})}\frac{\pi}{d}}.$$

For the outcome of each action, we replace τ with μ and we get

$$|\hat{\mu}(\boldsymbol{x}, a) - \mu(\boldsymbol{x}, a)| \leq \sigma_n(\boldsymbol{x})\delta + 2Md\left(\frac{\varepsilon s}{2k-1}\right)^{-\frac{1}{2}\frac{\log\left((1-\alpha)^{-1}\right)}{\log(\alpha^{-1})}\frac{\pi}{d}}. \quad (24)$$

In particular, with a probability at least $1 - e^{-\delta^2/2}$, the regret in round n for the online oracle

$$\begin{aligned}
&r_n = \mu(\boldsymbol{x}, a^*) - \mu(\boldsymbol{x}, a)\\
&= \left(\mu(\boldsymbol{x}, a^*) - \hat{\mu}(\boldsymbol{x}, a^*)\right) - \left(\mu(\boldsymbol{x}, a) - \hat{\mu}(\boldsymbol{x}, a)\right) + \left(\hat{\mu}(\boldsymbol{x}, a^*) - \hat{\mu}(\boldsymbol{x}, a)\right)\\
&\leq \left(\mu(\boldsymbol{x}, a^*) - \hat{\mu}(\boldsymbol{x}, a^*)\right) - \left(\mu(\boldsymbol{x}, a) - \hat{\mu}(\boldsymbol{x}, a)\right)\\
&\leq |\left(\mu(\boldsymbol{x}, a^*) - \hat{\mu}(\boldsymbol{x}, a^*)\right)| + |\left(\mu(\boldsymbol{x}, a) - \hat{\mu}(\boldsymbol{x}, a)\right)|\\
&\leq 2\sigma_n(\boldsymbol{x})\delta + 4Md\left(\frac{\varepsilon s}{2k-1}\right)^{-\frac{1}{2}\frac{\log\left((1-\alpha)^{-1}\right)}{\log(\alpha^{-1})}\frac{\pi}{d}}.
\end{aligned}$$

We let $\delta_0 = e^{-\delta^2/2}$, then $\delta = \sqrt{2\log(1/\delta_0)}$. Suppose $Var[T(x)]$ is uniformly bounded by V (which is in the theorem's condition), then

$$r_n \leq 2\sqrt{n^{\beta-1}V}\sqrt{2\log(\frac{1}{\delta_0})} + 4Md\left(\frac{\varepsilon s}{2k-1}\right)^{-\frac{1}{2}\frac{\log\left((1-\alpha)^{-1}\right)}{\log(\alpha^{-1})}\frac{\pi}{d}} + \varepsilon\Delta_{\max} \quad (25)$$

with probability at least $1 - \delta_0$, where $\Delta_{\max}$ denotes the maximum regret for choosing a sub-optimal action[3]. Recall that we denote $A = \frac{\log((1-\alpha)^{-1})\pi}{\log(\alpha^{-1})d}$. Now we denote $\epsilon_0 = -\frac{A}{2+3A}$, $\varepsilon = n^{\epsilon_0}$. One can check that $\beta = 1 - \frac{2A}{2+3A} = \frac{1-A\epsilon_0}{1+A}$. Then (25) can be rewritten as

$$r_n \leq \left(2\sqrt{V}\sqrt{2\log(\frac{1}{\delta_0})} + 4Md(2k-1)^{\frac{1}{2}A} + \Delta_{\max}\right)n^{-\frac{A}{2+3A}}. \quad (26)$$

Here, we notice $(\varepsilon s)^{-\frac{1}{2}A} = n^{-\frac{1}{2}A(\beta+\epsilon_0)}$. Thus we set the parameters so that each terms in (24) have the same exponent w.r.t. n:

$$\frac{1}{2}(\beta-1) = -\frac{1}{2}A(\beta+\epsilon_0) = \epsilon_0 = -\frac{A}{2+3A}$$

Let $C_3 = \left(2\sqrt{V}\sqrt{2\log(\frac{1}{\delta_0})} + 4Md(2k-1)^{\frac{1}{2}A} + \Delta_{\max}\right)$ be a constant. Then, we further denote $p \triangleq \frac{2+3A}{A} > 1$ and by Hölder's inequality,

$$\begin{aligned}
R(T, \mathcal{A}_5) = \sum_{n=1}^{T} r_n &\leq T^{1-1/p}C_3\left(\sum_{n=1}^{T}(\frac{r_n}{C_3})^p\right)^{1/p} = C_3T^{1-\frac{1}{p}}(\sum_{n=1}^{T}\frac{1}{n})^{\frac{1}{p}}\\
&\leq C_3T^{1-\frac{1}{p}}(\log n)^{\frac{1}{p}},
\end{aligned}$$

where the last inequality is because $\sum_{n=1}^{T}\frac{1}{n} \leq \log n$.

Now, we have the high-probability bound. We let $\delta_0 = T^{-1/p}$, then we have $\lim_{T\to+\infty} R(T, \mathcal{A}_5) = O(T^{1-\frac{A}{2+3A}+\xi})$ (or $\lim_{T\to+\infty}\frac{R(T,\mathcal{A}_5)}{T} = O(T^{-\frac{A}{2+3A}+\xi})$) for any small $\xi > 0$.

Finally, one can verify $1 - \frac{A}{2+3A} = \frac{2}{3-\beta}$. Then, we reach our claim in the theorem that $\lim_{T\to+\infty} R(T, \mathcal{A}_5)/T^{(2+\xi)/(3-\beta)} < +\infty$, and $\lim_{T\to+\infty} R(T, \mathcal{A}_5)/T = 0$. Namely, we have shown that the asymptotic regret is sub-linear which depends on the parameter α. □

Now, let us go back to our main proof. When in each time slot t, we have a probability ϵ_t to draw a random action, the average probability to randomly draw an action is comparable to the fixed rate ϵ. It is formalized by the following lemma.

Lemma 13. *We have the following bound for the sum of power*

$$T^{1-p} \leq \sum_{t=1}^{T} t^{-p} \leq T^{1-p}\log(T), \qquad \text{for some } p \in (0, 1).$$

[2]The proofs of paper[37] analyzes the estimation $\hat{\mu}(\boldsymbol{x}, a)$ directly, although their theorems' statements discuss the treatment effect.

[3]For $\Delta_{\max}$ to exist, we have a mild assumption that the average rewards are bounded for each actions.

Applying to our case, we let $p = -\epsilon_0$, and

$$T^{1+\epsilon_0} \leq \sum_{t=1}^{T} t^{\epsilon_0} \leq T^{1+\epsilon_0}\log(T).$$

Proof. The left inequality is easy to show. As t^{-p} decreases in t, $T^{-p} \leq t^{-p}$ for any $t \leq T$, and thus $\sum_{t=1}^{T} T^{-p} \leq \sum_{t=1}^{T} t^{-p}$. Now, we show the right inequality. According to Cauchy-Schwartz inequality (note that $1/p > 1$),

$$\frac{\sum_{t=1}^{T} t^{-p}}{T} \leq \left(\frac{\sum_{t=1}^{T}(t^{-p})^{1/p}}{T}\right)^{p}$$

$$= \left(\frac{\sum_{t=1}^{T} t^{-1}}{T}\right)^{p} \leq \left(\frac{\log(T)}{T}\right)^{p}.$$

Then, we get the right inequality $\sum_{t=1}^{T} t^{-p} \leq T^{1-p}\log(T)$. □

As a consequence, we can see by using the decreasing exploration rate ϵ_t, the expected number of times to do exploration is greater than that for the uniform exploration rate. On the other hand, the total cost of exploration increases less than a logarithmic factor $\log(T)$ (which is absorbed by any polynomial factor). Therefore, the bound in Lemma 11 for the fixed ϵ still holds for the case of time-decreasing $\{\epsilon_t\}_{t=1}^{T}$. Now, we complete the proof for the Theorem. □

A.4 Dropping the ignorability assumption

We also consider that the offline causal inference algorithms produce biased estimators. Suppose the offline causal inference algorithm has bias Δ because we do not have the propensity score (or the unconfoundedness condition).

Theorem 6 (No ignorability). *Assumptions 1, 2 hold. For a context-independent algorithm $\mathcal{A}$ using the UCB oracle, suppose the offline evaluator returns $\{y_j\}_{j=1}^{N}$ w.r.t. $\{(\boldsymbol{x}_j, a_j)\}_{j=1}^{N}$. The bias of the average outcome for action a is $\delta_a \triangleq (\sum_{j=1}^{N} \mathbb{1}_{\{a_j=a\}} y_j)/(\sum_{j=1}^{N} \mathbb{1}_{\{a_j=a\}}) - \mathbb{E}[y|a]$. Denote $N_a \triangleq \sum_{j=1}^{N} \mathbb{1}_{\{a_j=a\}}$, and recall a^* is the optimal action. Then,*

$$R(T, \mathcal{A}) \leq \sum_{a \neq a^*} \Delta_a \left(16\frac{\ln(N_a+T)}{\Delta_a^2} - 2N_a\left(1 - \frac{\max\{0, \delta_a - \delta_{a^*}\}}{\Delta_a}\right) + \left(1 + \frac{\pi^2}{3}\right)\right).$$

Proof. Let us consider the number of times that a sub-optimal action is played, using the UCB online bandit oracle. Let us denote the expected reward (or outcome) $\mathbb{E}[y|a]$ for an action a as μ_a. In the t_{th} online round, we make the wrong decision to play an action a only if $(\mu_{a^*} - \mu_a) + \left(\frac{\delta_{a^*} N_a}{N_a + t} - \frac{\delta_a N_a}{N_a + t}\right) < I_a - I_{a^*}$, where I_a is half of the width of the confidence interval $\beta\sqrt{\frac{2\ln(n)}{n_a}}$ for action a, where n_a is the number of times that the online bandit oracle plays action a and $n = \sum_{a \in [K]} n_a$. In the following, we only need to consider the case where $\delta_a - \delta_{a^*} \geq 0$. Otherwise, the offline data lets us to have less probability to select the sub-optimal action, and thus leads to a lower regret.

According to Chernoff bound, when we have

$$(N_a + t)[\Delta_a + \frac{N_a}{N_a + t}(\delta_{a^*} - \delta_a)]^2 \geq 8\ln(N_a + T), \qquad (27)$$

the violation probability will be very low. In fact, under (27)

$$\mathbb{P}\left[(\mu_{a^*} - \mu_a) + \left(\frac{\delta_{a^*} N_a}{N_a + t} - \frac{\delta_a N_a}{N_a + t}\right) < I_a - I_{a^*}\right] \leq t^{-4}.$$

Then we can let l_a to be a number such that when $t > l_a$, the inequality (27) is satisfied.

In fact, when we let $l_a = \lceil 16\frac{\ln(N_a+T)}{\Delta_a^2} + [N_a(\frac{2(\delta_a - \delta_{a^*})}{\Delta_a} - 1)] - N_a\rceil$, (27) is satisfied. Therefore, the expected number of times that we play an action a is less than

$$l_a + \sum_{t=1}^{T} t^{-4} \leq \left(16\frac{\ln(N_a+T)}{\Delta_a^2} - 2N_a\left(1 - \frac{\max\{0, \delta_a - \delta_{a^*}\}}{\Delta_a}\right) + \left(1 + \frac{\pi^2}{3}\right)\right).$$

□

A.5 More analysis whose theorems are not in the main paper

A.5.1 **Linear-model-based algorithms.** The following two theorems show more regret bounds on the linear-model-based algorithm $\mathcal{A}_4$, which serve as the supplement of our main paper. In particular, we consider the more general problem-independent bound in the following theorem.

Theorem 14 (Linear regression + LinUCB, problem-independent). *Suppose we have N offline data points. With probability at least $1 - \delta$, the psuedo-regret*

$$R(T, \mathcal{A}_4) \leq \sqrt{8(N+T)\beta_n(\delta)\log\frac{\texttt{trace}(V_0) + (N+T)L^2}{\texttt{det}V_0}}$$

$$- \sqrt{8\beta_n(\delta)}\min\{1, ||\boldsymbol{x}||_{\min}\}\frac{2}{L^2}\left(\sqrt{1 + NL^2} - 1\right).$$

Here, $\beta_t(\delta)$ is a non-decreasing sequence and $\beta_t(\delta) \geq 1$, and $L = ||\boldsymbol{x}||_{\max}$. One possible choice of β_t is $\beta_t = 2d(1 + 2\ln(1 + \delta))$.

Proof. The proof follows the analytical framework of the paper[1]. Especially, this Theorem corresponds to the Theorem 3 in the paper[1]. The proofs in papers[5][13] have similar ideas.

In particular, we consider that the offline samples have features $\boldsymbol{x}_{-1}, \boldsymbol{x}_{-2}, \ldots, \boldsymbol{x}_{-N}$, and the online samples have features $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^T$. To have a unified index system, we let $\boldsymbol{x}_{N+t} \triangleq \boldsymbol{x}^t$ for $t \geq 1$.

Because we choose the "optimal" action in the online phase, we have the pseudo-regret in time slot t is

$$r_t \leq 2\sqrt{\beta_{t-1}(\delta)}\min\{||\boldsymbol{x}_{N+t}||_{V_{N+t-1}^{-1}}, 1\}.$$

Then, we have

$$\sqrt{8\beta_n(\delta)}\sum_{n=1}^{N}\min\{1, ||\boldsymbol{x}_n||_{V_{n-1}^{-1}}\} + \sum_{t=1}^{T} r_t$$

$$\leq \sqrt{8(N+T)\beta_n(\delta)\log\frac{\texttt{trace}(V_0) + (N+T)L^2}{\texttt{det}V_0}}$$

Here, we observe that $\sum_{t=1}^{T} r_t \leq \sqrt{8(N+T)\beta_n(\delta)\log\frac{\texttt{trace}(V_0)+(N+T)L^2}{\texttt{det}V_0}} - \sqrt{8\beta_n(\delta)}\sum_{n=1}^{N}\min\{1, ||\boldsymbol{x}_n||_{V_{n-1}^{-1}}\}$. So we now give a lower bound of the last term $\sqrt{8\beta_n(\delta)}\sum_{n=1}^{N}\min\{1, ||\boldsymbol{x}_n||_{V_{n-1}^{-1}}\}$.

Here, $||\boldsymbol{x}||_A = \sqrt{\boldsymbol{x}^T A \boldsymbol{x}} \geq \sqrt{\lambda_{\min}(A)}||\boldsymbol{x}||_2$. We have the following claim that $\lambda_{\min}(V_n^{-1}) \geq \frac{1}{1+(n-1)L^2}$. This is because $\lambda_{\min}(V_n^{-1}) = 1/\lambda_{\max}(V_n)$. In fact, for the symmetric matrices, we have

$$\lambda_{\max}(A + B) \leq \lambda_{\max}(A) + \lambda_{\max}(B).$$

We have $\lambda_{\max}(I) = 1$, and $\lambda_{\max}(\boldsymbol{x}\boldsymbol{x}^T) = ||\boldsymbol{x}||_2^2$. Therefore, $\lambda_{\max}(V_{n-1}) \leq 1 + ||\boldsymbol{x}_1||_2^2 + \ldots + ||\boldsymbol{x}_{n-1}||_2^2 \leq 1 + (n-1)||\boldsymbol{x}||_{\max}^2$, where we consider $||\boldsymbol{x}_i||_2^2 \leq ||\boldsymbol{x}||_{\max}^2$ for $i \in [n]$. Also, we consider $||\boldsymbol{x}_i||_2^2 \geq ||\boldsymbol{x}||_{\min}^2$ for $i \in [n]$.

Let $L = ||\boldsymbol{x}||_{\max}$. Then,

$$\sum_{n=1}^{N} \min\{1, ||\boldsymbol{x}_n||_{V_{n-1}^{-1}}\} \geq \sum_{n=1}^{N} \min\{1, ||\boldsymbol{x}||_{\min}\sqrt{\frac{1}{1+(n-1)L^2}}\}$$

$$\geq \min\{1, ||\boldsymbol{x}||_{\min}\} \sum_{n=1}^{N} \sqrt{\frac{1}{1+(n-1)L^2}}$$

$$\geq \min\{1, ||\boldsymbol{x}||_{\min}\} \sum_{n=1}^{N} \frac{2}{L^2}\left(\sqrt{1+nL^2} - \sqrt{1+(n-1)L^2}\right)$$

$$= \min\{1, ||\boldsymbol{x}||_{\min}\} \frac{2}{L^2}\left(\sqrt{1+NL^2} - 1\right).$$

Hence, we have the final bound of regret

$$\sum_{t=1}^{T} r_t \leq \sqrt{8(N+T)\beta_n(\delta)\log\frac{\texttt{trace}(V_0)+(N+T)L^2}{\texttt{det}V_0}} - \sqrt{8\beta_n(\delta)}\min\{1, ||\boldsymbol{x}||_{\min}\}\frac{2}{L^2}\left(\sqrt{1+NL^2}-1\right).$$

□

Compared with the previous regret bound without offline data, the regret bound changes from $O(\sqrt{T})$ to $O(\sqrt{N+T})-\Omega(\sqrt{N})$. From the view of regret-bound, using offline data does not bring us a large amount of regret-reduction.

We now show a better bound for the problem-dependent case. This corresponds to section 5.2 of the paper[1]. Let Δ_t be the "gap" at step t as defined in the paper of Dani et al.[14]. Intuitively, Δ_t is the difference between the rewards of the best and the "second best" action in the decision set D_t. We consider the samllest gap $\bar{\Delta}_n = \min_{1\leq t\leq n}\Delta_t$.

Theorem 15. *Suppose the random error η_t is conditionally R-sub-Gaussian where $R \geq 0$ is a fixed constant. Assume that $\lambda \geq 1$ and $||\theta_*||2 \leq S$ where $S \geq 1$. With probability at least $1-\delta$, the regret*

$$\bar{R}(T) \leq \frac{4\beta_{N+T}(\delta)}{\bar{\Delta}}\left(\log\frac{\texttt{trace}(V_0)+(N+T)L^2}{\texttt{det}V_0} - \frac{||\boldsymbol{x}||_{\min}^2}{L^2}\log(1+(N-1)L^2)\right)$$

Proof. We again consider to add extra terms related to $||\boldsymbol{x}_n||_{V_{n-1}^{-1}}$.

$$\sum_{n=1}^{N}\frac{4\beta_{N+T}(\delta)}{\bar{\Delta}}||\boldsymbol{x}_n||^2_{V_{t-1}^{-1}} + \sum_{t=1}^{T} r_t \leq 8\beta_{N+T}(\delta)\log\frac{\texttt{det}V_{N+T}}{\texttt{det}V_0}$$

$$\leq \frac{4\beta_{N+T}(\delta)}{\bar{\Delta}} d\log\frac{v_0+(N+T)L^2}{d\texttt{det}^{1/d}V_0}$$

We also have the bound that

$$\sum_{n=1}^{N}||\boldsymbol{x}_n||^2_{V_{t-1}^{-1}} \geq \sum_{n=1}^{N}||\boldsymbol{x}||^2_{\min}\frac{1}{1+(n-1)L^2}$$

$$> ||\boldsymbol{x}||^2_{\min}\int_1^{N+1}\frac{1}{1+(x-1)L^2}dx = \frac{||\boldsymbol{x}||^2_{\min}}{L^2}\log(1+(N-1)L^2).$$

Therefore, the pseudo-regret has the bound

$$\sum_{t=1}^{T} r_t \leq \frac{4\beta_{N+T}(\delta)}{\bar{\Delta}}\left(\log\frac{\texttt{trace}(V_0)+(N+T)L^2}{\texttt{det}V_0} - \frac{||\boldsymbol{x}||^2_{\min}}{L^2}\log(1+(N-1)L^2)\right)$$

□

When $||\boldsymbol{X}||_{\min} = L$, the pseudo-regret is $O(\log(T))$ when the number of offline data N is proportional to the number of online rounds, which is better than the previous upper bound $O(\log)$.

A.5.2 **Causal forest algorithms.** Then, for the special case where the structure the of random forest is known, we analyze the regret. This case could be regarded as the parametric case because the structure of the tree is already known and the only unknown part is the parameters in the forest. Then, the problem reduces to a parametric-estimation problem. Suppose there are B trees in the random forest.

Definition 1. *We say the forest estimator $\{T_1, \ldots, T_B\}$ is unbiased, if for each possible context and action $\boldsymbol{x} \in \mathcal{X}$ and $a \in [K]$, we have $\mathbb{E}[f(x,a)] = \frac{1}{B}\sum_{b=1}^{B} T_b(\boldsymbol{x}, a)$.*

Theorem 16 (ϵ-greedy causal forest, problem independent). *Suppose $Y_i \in [0,1]$ is bounded. For the forest estimator $\{T_1, \ldots, T_B\}$ which is unbiased, we have the following regret bound.*

$$\bar{R}_T \leq \sqrt{\frac{2}{B}}\sqrt{\log(\frac{1}{\delta})T\sum_{b}^{B} K|\mathcal{L}_b|\,(1+\log(T))}$$

Proof. Let us use $N^t(L_b(\boldsymbol{x}), a)$ to denote the number of times that the leaf $L(\boldsymbol{x})$ and action a is selected in the tree T_b. The estimator regarding to the samples is

$$\hat{\mu}(x,a) = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{|\{i: A_i = a, X_i \in L_b(x)\}|}\sum_{\{i:A_i=a, X_i\in L_b(x)\}} Y_i.$$

One can see that the estimated outcome is a weighted sum of the samples. Here, the weight of sample i is $\frac{1}{B}\sum_{b=1}^{B}\mathbb{1}_{\{A_i=a, X_i\in L_b(x)\}}\frac{1}{N(L_b(\boldsymbol{x}),a)}$. The sample value is a function of the context $\boldsymbol{x}$ and the action a, and therefore each sample values are independent given the $\boldsymbol{x}$ and a. Then, by Hoeffding bound (we have n offline samples)

$$\mathbb{P}[|\mathbb{E}[\hat{\mu}(\boldsymbol{x},a)] - \hat{\mu}(\boldsymbol{x},a)| \geq \varepsilon] \leq e^{-\frac{2\varepsilon^2}{\sum_{i=1}^{n}\left(\frac{1}{B}\sum_{b=1}^{B}\mathbb{1}_{\{A_i=a, X_i\in L_b(x)\}}\frac{1}{N(L_b(\boldsymbol{x}),a)}\right)^2}}.$$

Let $\delta = e^{-\frac{2\varepsilon_t^2}{\sum_{i=1}^{n}\left(\frac{1}{B}\sum_{b=1}^{B}\mathbb{1}_{\{A_i=a_t, X_i\in L_b(x_t)\}}\frac{1}{N_t(L_b(\boldsymbol{x}_t),a_t)}\right)^2}}$, then

$$\varepsilon_t = \frac{1}{B}\sqrt{\log(\frac{1}{\delta})\frac{\sum_{i=1}^{n}\left(\sum_{b=1}^{B}\mathbb{1}_{\{A_i=a_t, X_i\in L_b(x_t)\}}\frac{1}{N_t(L_b(\boldsymbol{x}_t),a_t)}\right)^2}{2}}.$$

It means that with probability at least $1-\delta$, the regret in time t is less than $r_t \leq 2\varepsilon_t$.

The total regret

$$\sum_{t=1}^{T} r_t \leq \sqrt{T\sum_{t=1}^{T} r_t^2}$$

$$= \frac{2}{B}\sqrt{T\sum_{t=1}^{T}\log(\frac{1}{\delta})\frac{\sum_{i=1}^{t-1}\left(\sum_{b=1}^{B}\mathbb{1}_{\{A_i=a_t, X_i\in L_b(\boldsymbol{x}_t)\}}\frac{1}{N_t(L_b(\boldsymbol{x}_t),a_t)}\right)^2}{2}}$$

$$\leq \frac{2}{B}\sqrt{T\sum_{t=1}^{T}\log(\frac{1}{\delta})\frac{\sum_{i=1}^{t-1}B\sum_{b=1}^{B}\left(\mathbb{1}_{\{A_i=a_t, X_i\in L_b(\boldsymbol{x}_t)\}}\frac{1}{N_t(L_b(\boldsymbol{x}_t),a_t)}\right)^2}{2}}$$

$$= \sqrt{\frac{2}{B}}\sqrt{T\sum_{t=1}^{T}\log(\frac{1}{\delta})\sum_{b=1}^{B}\sum_{i=1}^{t-1}\left(\mathbb{1}_{\{A_i=a_t, X_i\in L_b(\boldsymbol{x}_t)\}}\frac{1}{N_t(L_b(\boldsymbol{x}_t),a_t)}\right)^2}$$

Let us now consider the specific quantity

$$\sum_{t=1}^{T}\sum_{i=1}^{t-1}\left(\mathbb{1}_{\{A_i=a_t, X_i\in L_b(\boldsymbol{x}_t)\}}\frac{1}{N_t(L_b(\boldsymbol{x}_t),a_t)}\right)^2.$$

Note that $N_t(L_b(\boldsymbol{x}_t), a_t) = \sum_{i=1}^{t-1} \mathbb{1}_{\{A_i=a_t, X_i \in L_b(\boldsymbol{x}_t)\}}$, so

$$\sum_{t=1}^{T}\sum_{i=1}^{t-1}\left(\mathbb{1}_{\{A_i=a_t, X_i\in L_b(\boldsymbol{x}_t)\}}\frac{1}{N_t(L_b(\boldsymbol{x}_t),a_t)}\right)^2$$

$$=\sum_{t=1}^{T}\frac{1}{N_t(L_b(\boldsymbol{x}_t),a_t)} = \sum_{L\in\mathcal{L}_b}\sum_{a\in[K]}\left(1+\frac{1}{2}+\cdots+\frac{1}{N_T(L,a)}\right)$$

$$\leq \sum_{L\in\mathcal{L}_b}\sum_{a\in[K]}(1+\log(N_T(L,a))) \leq \sum_{L\in\mathcal{L}_b}\sum_{a\in[K]}(1+\log(T))$$

$$=K|\mathcal{L}_b|\,(1+\log(T))\,.$$

Therefore, the regret

$$\sum_{t=1}^{T} r_t \leq \sqrt{\frac{2}{B}}\sqrt{\log(\frac{1}{\delta})T\sum_{b}^{B}K|\mathcal{L}_b|\,(1+\log(T))}.$$

Let $\delta = 1/T$, then we have the problem independent cumulative regret is of order $O(\sqrt{T}\log T)$. Surely, the regret bound also depends on how fast the bias term decreas. □

Theorem 17 (UCB causal forest, problem-dependent). *Suppose the minimum reward gap is $\Delta_{\min}$, then with probability $1-\delta$ the cumulative regret*

$$\sum_{t=1}^{T} r_t \leq \frac{2}{B\Delta_{\min}^2}\log(\frac{1}{\delta})K|\mathcal{L}_b|\frac{\pi^2}{6}.$$

Specailly, when we set $\delta = 1/T$, the

$$R(T,\mathcal{A}_5) \leq \frac{2}{B\Delta_{\min}^2}\log(T)K|\mathcal{L}_b|\frac{\pi^2}{6}.$$

Proof. The proof is similar to the proof of the previous theorem.

We've tried to emulate the finite-time analysis on the UCB algorithm. The problem is that the confidence bound depends on more than one counters. Therefore, one cannot use the value of one counter as the threshold to decide whether we should use the tail probability.

Maybe we can try this direction, using the property that when $2\varepsilon \leq \Delta_{\min}$, the error is 0.

We have

$$\sum_{t=1}^{T} r_t \leq \sum_{t=1}^{T}\frac{r_t^3}{\Delta_{\min}^2}$$

$$\leq \frac{2}{B\Delta_{\min}^2}\sum_{t=1}^{T}\log(\frac{1}{\delta})\sum_{b=1}^{B}\sum_{i=1}^{t-1}\left(\mathbb{1}_{\{A_i=a_t, X_i\in L_b(\boldsymbol{x}_t)\}}\frac{1}{N_t(L_b(\boldsymbol{x}_t),a_t)}\right)^3$$

$$(\leq 1+\frac{1}{2^2}+\ldots+\frac{1}{N_T^2(L,a)} \leq \frac{\pi^2}{6})$$

$$\leq \frac{2}{B\Delta_{\min}^2}\log(\frac{1}{\delta})K|\mathcal{L}_b|\frac{\pi^2}{6}.$$

□

Theorem 18. *Suppose using the offline data, we have the initialization such that we have $\tilde{N}(L_b,a)$ data points for the tree leaf L_b in the tree b and action a. Then, we have*

$$\sum_{t=1}^{T} r_t \leq \frac{2}{B\Delta_{\min}}\log(\frac{1}{\delta})\sum_{L_b\in\mathcal{L}_b}\sum_{a\in[K]}(1+\log(\frac{T+\tilde{N}(L_b,a)}{\tilde{N}(L_b,a)})).$$

Proof. We just use the previous proof on the regret bound for the structured-forest. The only difference is that for the $\sum_{t=1}^{T}\frac{1}{N_t(L_b(\boldsymbol{x}_t),a_t)}$ term, we start from $\tilde{N}(L_b,a)$ and end before $T+\tilde{N}(L_b,a)$. □

We can see that the regret bound for the structured-forest algorithm is similar to the regret bound for the contextual linear bandit problem.

REFERENCES

[1] Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems* (2011), pp. 2312–2320.

[2] Agrawal, S., and Goyal, N. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory* (2012), pp. 39–1.

[3] Anonymous. Code and technical report. link: https://1drv.ms/u/s!AuhX-fJM-sJvgnXKSvpmReREQUe9?e=97NeFu.

[4] Athey, S., Tibshirani, J., Wager, S., et al. Generalized random forests. *The Annals of Statistics 47*, 2 (2019), 1148–1178.

[5] Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research 3*, Nov (2002), 397–422.

[6] Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time analysis of the multi-armed bandit problem. *Machine learning 47*, 2-3 (2002), 235–256.

[7] Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM journal on computing 32*, 1 (2002), 48–77.

[8] Austin, P. C. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate behavioral research 46*, 3 (2011), 399–424.

[9] Bareinboim, E., Forney, A., and Pearl, J. Bandits with unobserved confounders: A causal approach. In *Advances in Neural Information Processing Systems* (2015), pp. 1342–1350.

[10] Bertrand, M., Duflo, E., and Mullainathan, S. How much should we trust differences-in-differences estimates? *The Quarterly journal of economics 119*, 1 (2004), 249–275.

[11] Bottou, L., Peters, J., Quiñonero-Candela, J., Charles, D. X., Chickering, D. M., Portugaly, E., Ray, D., Simard, P., and Snelson, E. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research 14*, 1 (2013), 3207–3260.

[12] Bubeck, S., Perchet, V., and Rigollet, P. Bounded regret in stochastic multi-armed bandits. In *Conference on Learning Theory* (2013), pp. 122–134.

[13] Chu, W., Li, L., Reyzin, L., and Schapire, R. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (2011), pp. 208–214.

[14] Dani, V., Hayes, T. P., and Kakade, S. M. Stochastic linear optimization under bandit feedback. In *COLT* (2008).

[15] Dimakopoulou, M., Athey, S., and Imbens, G. Estimation considerations in contextual bandits. *arXiv preprint arXiv:1711.07077* (2017).

[16] Dong, S., and Van Roy, B. An information-theoretic analysis for thompson sampling with many actions. In *Advances in Neural Information Processing Systems* (2018), pp. 4157–4165.

[17] Dudík, M., Langford, J., and Li, L. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601* (2011).

[18] Féraud, R., Allesiardo, R., Urvoy, T., and Clérot, F. Random forest for the contextual bandit problem. In *Artificial Intelligence and Statistics* (2016), pp. 93–101.

[19] Forney, A., Pearl, J., and Bareinboim, E. Counterfactual data-fusion for online reinforcement learners. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), JMLR. org, pp. 1156–1164.

[20] Guan, M. Y., and Jiang, H. Nonparametric stochastic contextual bandits. In *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).

[21] Hansen, L. P. Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the Econometric Society* (1982), 1029–1054.

[22] Hoeffding, W. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*. Springer, 1994, pp. 409–426.

[23] Kallus, N. Balanced policy evaluation and learning. In *Advances in Neural Information Processing Systems* (2018), pp. 8895–8906.

[24] Kuleshov, V., and Precup, D. Algorithms for multi-armed bandit problems. *arXiv preprint arXiv:1402.6028* (2014).

[25] Li, L. Offline evaluation and optimization for interactive systems.

[26] Li, L., Chu, W., Langford, J., Moon, T., and Wang, X. 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* (2012), pp. 19–36.

[27] Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web* (2010), ACM, pp. 661–670.

[28] McCaffrey, D. F., Ridgeway, G., and Morral, A. R. Propensity score estimation with boosted regression for evaluating causal effects in observational studies. *Psychological methods 9*, 4 (2004), 403.
[29] Pearl, J. *Causality: models, reasoning and inference*, vol. 29. Springer, 2000.
[30] Pearl, J., and Mackenzie, D. *The book of why: the new science of cause and effect*. Basic Books, 2018.
[31] Rosenbaum, P. R., and Rubin, D. B. The central role of the propensity score in observational studies for causal effects. *Biometrika 70*, 1 (1983), 41–55.
[32] Rubin, D. B. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association 100*, 469 (2005), 322–331.
[33] Shivaswamy, P., and Joachims, T. Multi-armed bandit problems with history. In *Artificial Intelligence and Statistics* (2012), pp. 1046–1054.
[34] Stuart, E. A. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics 25*, 1 (2010), 1.
[35] Swaminathan, A., and Joachims, T. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning* (2015), pp. 814–823.
[36] Villar, S. S., Bowden, J., and Wason, J. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics 30*, 2 (2015), 199.
[37] Wager, S., and Athey, S. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association 113*, 523 (2018), 1228–1242.
[38] Wang, Y., Liang, D., Charlin, L., and Blei, D. M. The deconfounded recommender: A causal inference approach to recommendation. *arXiv preprint arXiv:1808.06581* (2018).
[39] Xu, Y., Chen, N., Fernandez, A., Sinno, O., and Bhasin, A. From infrastructure to culture: A/b testing challenges in large scale social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), ACM, pp. 2227–2236.
[40] Zhang, C., Agarwal, A., Iii, H. D., Langford, J., and Negahban, S. Warm-starting contextual bandits: Robustly combining supervised and bandit feedback. In *International Conference on Machine Learning* (2019), pp. 7335–7344.