Incrementality Bidding via Reinforcement Learning under Mixed and Delayed Rewards

Anonymous Author(s)
Affiliation
Address
email

Abstract

Incrementality, which is used to measure the causal effect of showing an ad to a potential customer (e.g. a user in an internet platform) versus not, is a central object for advertisers in online advertising platforms. This paper investigates the problem of how an advertiser can learn to optimize the bidding sequence in an online manner *without* knowing the incrementality parameters in advance. We formulate the offline version of this problem as a specially structured episodic Markov Decision Process (MDP) and then, for its online learning counterpart, propose a novel reinforcement learning (RL) algorithm with regret at most $\widetilde{O}(H^2\sqrt{T})$, which depends on the number of rounds H and number of episodes T, but does not depend on the number of actions (i.e., possible bids). A fundamental difference between our learning problem from standard RL problems is that the realized reward feedback from conversion incrementality is *mixed* and *delayed*. To handle this difficulty we propose and analyze a novel pairwise moment-matching algorithm to learn the conversion incrementality, which we believe is of independent of interest.

1 Introduction

Nowadays, online advertising systems (e.g. Google, Facebook, and Amazon) have demonstrated their significant power for connecting advertisers and potential customers in practice. Moreover, automated auctions are widely used in online ad platforms for matching advertisers and users, and for price discovery [22, 8]. Many advertisers are bidding repeatedly online to show ads to users. Therefore, an important problem for advertisers is to design efficient bidding algorithms to maximize their accumulated utility. Moreover, this is also important for the online ad platforms, because *auto-bidding*— i.e., the advertisers authorize ad platforms to bid on behalf of them —has become prevalent in online advertising, e.g., [9, 11].

A growing body of work is investigating the problem of *learning to bid* in repeated auctions [23, 10, 5, 13, 12, 4, 19, 18]. However, existing work assumes that the value of showing ads to a user (after winning the auction) comes from either an oblivious adversary or is sampled from an underlying fixed distribution. Unfortunately, this ignored the crucial *causal effect* of showing an ad to a potential customer and cannot capture how much an ad will *change* the user's conversion rate, which should have been what the advertiser is bidding for. For example, if a user sees the same ad multiple times in a very short time period, then the conversion of this user may not be much better — in fact, may even be worse — than that of showing ads to this user only once within this period. Such multiple redundant ad placements are commonly known in reality to be cost-inefficient for the advertisers, but cannot be captured by previous models of learning to bid with either random or completely adversarial ad values. To overcome the drawbacks of previous approaches to measuring the value of an advertising opportunity, a recent visionary whitepaper by Lewis and Wong [17] introduced the important notion of *incrementality*, which properly quantifies this causal effect of showing an ad to

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

the user as validated by massive real data from leading advertising platforms. However, [17] focuses on the offline learning problem of estimating incrementality from past data. The online learning of incrementality parameters, as well as how to utilize it to optimize bidding sequences, has not been studied; this is what we embark on in this paper.

In Figure 1, we visualize the conversion incrementality caused by three ad placements and used the colored areas to capture the incrementality caused by each ad placement. There are two main challenges in learning incrementality, i.e., *delayed* and *mixed* conversion feedback. First, the conversion may not happen immediately when the ad is shown to the user, even though the conversion rate may quickly peak after showing the ad (see Ad 2 and Ad 3 in Figure 1). Such delayed conversion feedback in online advertising is studied only recently in offline setups, e.g., [6, 20, 21, 3]. Second, when a conversion happens, it is a mixed effect from multiple previous ad placements before the conversion. In this work, we formulate incrementality bidding as an episodic Markov Decision Process (MDP), where each episode represents an interaction with a single user drawn from a population of the same characteristics (and thus assumed to have the same incrementality parameters). The advertiser wants to learn the incrementality and optimize bidding for this user population. This is hardly possible in general given the two difficulties above as well as the potentially complex dependence of incrementality on the entire history. We thus make a Markovian assumption that the conversion incrementality of ad placement at each round h only depends on its *last* ad impression shown before this round in the current episode. In addition, we introduce the *heterogeneous Poisson process* to formally capture the mixed and delayed rewards feedback in the conversion process (Poisson process is also mentioned by Lewis and Wong [17], but only qualitatively).

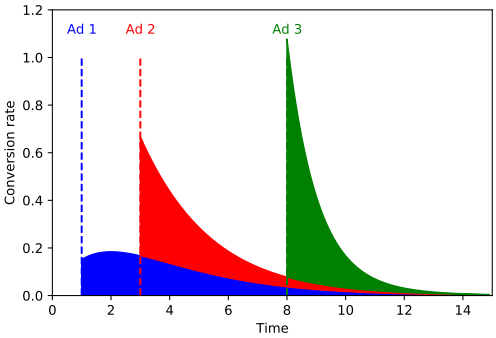

Figure 1: Toy example: visualization of the conversion incrementality triggered by three ad impressions. The rate of conversion incrementality of each ad follows $\texttt{Gamma}(2, 2)$, $\texttt{Exp}(0.5)$ (shifted right by 3) and $\texttt{Exp}(1)$ (shifted right by 8) distributions, respectively.

To optimize the sequential bidding strategies in the online setup, there is an intrinsic exploration versus exploitation tradeoff in incrementality bidding. The advertiser needs to win to place the ad to learn the conversion incrementality, however, she also doesn't want to bid too high so that the conversion reward cannot cover her cost (payment to win the auction to show the ad). This observation naturally motivates our design of a model-based RL algorithm that adopts a UCB-style approach to balance exploration and exploitation.[1]

Our Results and Contributions. The main contributions of this paper are two folds: (1) we introduce an episodic MDP, together with a heterogeneous Poisson process, to capture the incrementality bidding problem; (2) we design a novel reinforcement learning algorithm for the advertiser to optimize her sequential bidding while learning incrementality simultaneously. *Technically*, our main result is the design of an RL algorithm for incrementality bidding that provably has regret at most $\widetilde{O}(H^2\sqrt{T})$ and is near-optimal in terms of the dependence on T. *Conceptually*, our result demonstrates the possibility of designing highly-efficient RL algorithms for incrementality bidding despite its various challenges. To the best of our knowledge, this is the first work that analyzes incrementality bidding in an online setting theoretically.

Notably, our regret bound is independent of the number of possible bids (equiv. actions in standard RL). This is due to the special structures of the incrementality bidding problem, which has a small *effective* action space. Specifically, there are only two effective outcomes for an advertiser—winning or losing— that eventually affect the state transitions. The conversion incrementality only happens if the learner wins. Finally, we remark that our results are independent of auction formats and all our results hold for second price auctions, first price auctions, and other standard single-item auction

[1] We remark that designing model-free no-regret RL algorithms is an intriguing open problem. Due to delayed and mixed reward feedback, it appears challenging to use algorithms like Q-learning in the incrementality bidding problem.

formats. To achieve this regret bound, we provide a novel regret decomposition to incorporate with the convergence rate analysis of PAMM method.

One key technical novelty of this paper is a new estimation method, *Pairwise Moment-Matching* (PAMM) algorithm, that can estimate reward parameters under mixed and delayed conversion feedback. In this type of models with mixture effects, standard likelihood-based model fitting procedure would lead to solving non-convex optimization problems resulting in difficulties for computation and analysis, as well as unpleasant gap between theory and algorithm [15]. However, PAMM estimates incrementality parameters in an online and computationally efficient manner without the need of solving nonconvex optimization problems. The estimators are provably consistent with a nearly-optimal convergence rate. We believe this novel technique will be of independent interest for other online learning problems with delayed and mixed reward feedback.

Related Work. As mentioned in the introduction, our work is highly inspired by the visionary work of Lewis and Wong [17], who firstly proposed incrementality bidding (and attribution). However, the method proposed in [17] lacks a theoretical guarantee and cannot be adapted to online learning settings. Our work is generally related to the *learning to bid* literature [23, 10, 5, 13, 12, 4]. However, all these existing papers usually model this problem as (contextual) bandits and don't consider the causal effect of showing an ad to the user captured by incrementality (i.e., the additional value or conversion given the ad placements in previous rounds), which causes the main difference from this paper. Specifically, in all these previous works [23, 10, 5, 13, 12, 4], the reward for an ad impression is realized immediately at the current round whereas, in our model, any ad impression only increases the rate of conversions, which however may be realized in any later time following a Poisson process and, more importantly, is mixed with the incrementality of other ad impressions. We believe our model captures the effect of advertising more realistically. Some other less related works include delayed feedback in other different RL problems, which have only been studied recently [16, 14]. We handle delayed conversion feedback through a parametric inhomogeneous Poisson process, which is different from these works.

2 The Sequential Optimization Problem of Incrementality Bidding (IB)

This paper adopts the perspective of an online advertiser, and studies the advertiser's learning problem of optimizing her sequential bidding policy. Next, we will first describe the background and motivation of this problem and then introduce the formal model.

We consider the sequential bidding problem for a single advertiser — i.e., the *decision maker* or *learner* — in online advertising systems. The advertiser participates online advertising auctions in order to win opportunities for showing ads to Internet users and, ultimately, gain conversions (i.e., purchases of products or services) from Internet users. The interaction between the advertiser and Internet users happens in an *episodic* manner; each episode corresponds to one Internet user. Specifically, each episode has H rounds, during which the advertiser (learner) will interact with the user of that episode for at most H times. Notably, H is a finite number since (1) an interaction happens only when a user visits a particular webpage and (2) if an advertiser already bids for a user for sufficiently many times and still was not able to get conversion, the advertiser typically will cease to advertise to this user (at least for some time). We use T to denote the total number of episodes.[2] To focus on the fundamentals of the problem, this work studies the basic setup with T i.i.d. episodes (i.e., Internet users) that occur *sequentially* and evenly distributed H rounds at integer time point $1, 2, \cdots, H$ within each episode. Interesting future directions include examining the more general settings in which each Internet user in the corresponding episode may depend on a context feature, the time points of the H rounds are different for different episodes, and the episodes may happen in batch (i.e., multiple similar users arrive at the same time).

Conversion incrementality. Central to our bidding optimization problem is the notion of *"conversion incrementality"* [17]. Intuitively, incrementality captures how much *increase* the advertiser's ads can boost the user's conversion rates. Notably, the reward of showing an ad is the incrementality while not the conversion rate itself. This is very different from the previous *learning to bid* literature since they assume the conversion (or value) comes independently with the previous ad placements

[2] In reality, H is usually at the scale of around 50 (the number of times an advertiser would like to interact with a user), whereas the number of similar Internet users (i.e., episodes) T is of the scale of millions.

(i.e., from oblivious adversary or an underlying distribution). This is why previous studies almost all use the (contextual) bandits setup, while not MDP.

We now formally describe the model of incrementality bidding, following [17]. Within any episode $t \in [T]$, each round $h \in [H]$ has an *incrementality rate function* $d_h(\tau; A_h, \theta_h)$ for any continuous time $\tau \in [h, \infty)$ which is the "density" of the conservation incrementality at any time $\tau \in [h, \infty)$. Intuitively, $d_h(\tau; A_h, \theta_h)$ models how much an ad shown at round h boosts the conversion rate density at any future time $\tau > h$. Notably, the integral of this rate function $d_h(\tau; A_h, \theta_h)$ does not need to be 1 as there may be many conversions per user impression. We assume that $d_h(\tau; A_h, \theta_h)$ depends on an to-be-learnt unknown parameter θ_h and additionally a subset $A_h \subset [h]$ containing all previous rounds, at which our learner's ads were shown to the corresponding user of this episode. Let $\theta = (\theta_1, \theta_2, \cdots, \theta_H)$ denote all the to-be-learnt parameters.

It is generally intractable to learn an arbitrary rate function $d_h(\tau; A_h, \theta_h)$. Next, we introduce a natural *parameterized family* of functions for $d_h(\tau; A_h, \theta_h)$, with to-be-learnt parameters, in order to capture advertising applications. First, we have the following assumption for d_h,

Assumption 1 (Markovian Incrementality)**.** *d_h is only affected by the learner's* last *ad impression before h, rather than the entire history A_h of the learner's advertising performance. Formally, at current round h, suppose the last round at which the learner's ad was shown is $h - l$, then the incrementality function $d_h(\tau; A_h, \theta_h) = d_h(\tau; \ell, \theta_h)$.*

The above assumption is built upon the sense that the users normally weighs more on the recently visited ads and are generally memory-less. Moreover, we assume each incrementality rate function $d_h(\tau; \ell, \theta_h)$ for any $h \in [H]$ has the following parametric form:

$$d_h(\tau; \ell, \theta_h) = \begin{cases} \beta_h(\ell)\lambda_h e^{-(\tau-h)\lambda_h} & \forall \tau \in [h, \infty), \\ 0 & otherwise, \end{cases} \tag{1}$$

in which $\theta_h = \{\beta_h(\ell)\}_{\ell=1}^{h-1} \cup \{\lambda_h\}$ contain all the parameters at round h of the given `IB` instance. In other words, the incrementality rate is a re-scaled exponential distributions and the re-scaling is due to fact that each user impression may lead to more (or less) than one user conversions. In our RL formulation of the problem, these are all the to-be-learnt parameters. The Markovian property is reflected in $\beta_h(\ell)$. Note that $\int_{\tau=h}^{\infty} d_h(\tau; \ell, \theta_h) d\tau = \beta_h(\ell)$. So $\beta_h(\ell)$ can be viewed as the expected number of conversions triggered by the ad impression at round h, given the last time for showing the learner's ad is ℓ rounds before. In this paper, we assume $\forall h, \ell, c_\beta \leq \beta_h(\ell) \leq C_\beta$ for some positive constants c_β and C_β. Without loss of generality[3], we assume $C_\beta \leq 1$.

Remark 1. *The assumption of exponential decaying rate in Equation* (1) *is not necessary for our approach to work, and is primarily for the exposition. Our techniques can be applied to most parametric family of rate functions such as truncated exponential, logistic, Beta, and Gamma distributions [17], as long as the CDF function is invertible.*

From conversion incrementality to realized conversions. While incrementality captures how the rate of conversions increases, it did not model how conversions are realized in reality. This would be important when we introduce our reinforcement learning problem — after all, in real applications, the only observations our learner (an advertiser) can see are the realized conversions at different time, rather than the continuous conversion rate function. To capture this, we adopt the standard assumption that conversions are realized based on a continuous-time inhomogeneous *Poisson process* [17]. More specifically, let the *ordered* subset $W = \{w_1, w_2, \cdots, w_n\} \subseteq [H]$ denote the rounds at which the learning agents win (for convenience, let $w_0 = 0$) in one episode. Then the conversions arrive according to a Poisson process with a time-varying rate $r(\tau)$, defined as follows

$$r(\tau; w, \theta) = \sum_{i=1}^{n} d_{w_i}(\tau; w_i - w_{i-1}, \theta_{w_i}), \qquad \forall \tau \in [0, H]. \tag{2}$$

More details of inhomogeneous Poisson process is deferred to Appendix A.

The optimization problem of bidding for incrementality. While the above incrementality modeling of user conversions originates from the influential idea of [17], our following formulation of the

[3]Our results can be directly applied to other conversion rate functions as long as $\beta_h(\ell)$ is bounded $\forall h, \ell$.

optimal bidding problem under their user modeling is new to the best of our knowledge, and so is its reinforcement learning solutions presented in this paper for addressing the situations with unknown environment parameters.

Following current practice, we assume online ad opportunities are sold to advertisers by auctions, and our learner is one of these advertisers. For the ease of presentation, we restrict our descriptions to the widely adopted *second price auctions*; however, all our results can be easily adapted to any single-item auction format without entry fees (e.g., the first price auction). Let $\mathcal{B}$ denote the bidding space, which may be continuous. Notably, $0 \in \mathcal{B}$, meaning the bidder/learner can choose to not participate this round's auction. Moreover, denote the distribution of the highest bid among *other* bidders (HOB) at round h as F_h (the cumulative distribution function or CDF). For any episode t (corresponding to an Internet user), at each round h, the learner submits a bid $b \in \mathcal{B}$ and wins the opportunity to show its ad to the user if her bid is the highest. Let $p_h : \mathcal{B} \to \mathbb{R}_{\geq 0}$ represents the average (conditional) payment function of the learner if she wins. Let v denote the learner's average value per conversion, which is known to the learner. We assume v is bounded, w.l.o.g., within $[0, 1]$. Then for any bid $b \in \mathcal{B}$, given the current state ℓ, the immediate expected utility of the learner at round h (for any episode) is,

$$u_h(b; \ell, \theta_h) = F_h(b) \cdot \left(\int_h^\infty v d_h(\tau; \ell, \theta_h) d\tau - p_h(b) \right) = F_h(b) \cdot (\beta_h(\ell) \cdot v - p_h(b)) \tag{3}$$

The astute reader may already find the expected utility function of the learner doesn't depend on parameter λ_h. This raises the question why we still need to know the function class of d_h (i.e., exponential function) and why we still need to learn parameter λ. As we will see later in Section 4, due to the complexity of inhomogeneous Poisson process, it is crucial to know the function class of d_h and we have to learn parameters λ first in order to get a good estimator to β afterwards. In second price auctions, $p_h(b)$ is the expected second highest bid conditioning on winning, which can be computed as follows

$$p_h(b) = \frac{1}{F_h(b)} \int_0^b v dF_h(v) = b - \frac{1}{F_h(b)} \int_0^b F_h(v) dv. \tag{4}$$

As mentioned above, our techniques apply to other auction formats as well. For example, in first price auctions, $p_h(b) = b$ has an even simpler format.

3 The Optimal Policy of `IB` and its Reinforcement Learning

While it may not appear obvious at the first glance, we explain how `IB` can be reduced to a finite-horizon Markov Decision Process (MDP) in this section and then introduce the corresponding reinforcement learning problem for `IB` when the environment parameters — namely, the incrementality parameters $\theta_h = \{\beta_h(l)\}_{l=1}^{h-1} \cup \{\lambda_h\}$ and HOB distribution F_h — are unknown.

3.1 The Offline Optimal Policy of `IB`

We now formulate the optimal planning problem for the `IB` problem in its offline version, i.e., when all environment parameters are known. We start by defining what is a policy for the `IB` problem.

Definition 1 (Policy)**.** *A deterministic (dynamic) policy is a sequence of mappings $\pi = (\pi_1, \cdots, \pi_H)$, in which mapping $\pi_h : [h] \to \mathcal{B}$ maps any $\ell \in [h]$ to a bid. Here, ℓ is chosen such that the last winning of the learner happens at round $h - \ell$.*

Note that it suffice to consider policies that depend only on the number of consecutive loses ℓ due to the Markovian property of incrementality as mentioned above.

The MDP re-formulation of `IB`. The MDP will have H *states*, with *state* $\ell \in [H]$ denoting that the last winning round is ℓ rounds ago from current round. The *action* set is $\mathcal{B}$, containing all possible bids. Both the state and action space are the same for each round h, however the transition and rewards are different for different rounds. Specifically, at round h, the transition probability from state l and action b is to enter state 1 at round $h + 1$ with probability $F_h(b)$, i.e., the probability of winning the auction, and to enter state $\ell + 1$ with probability $1 - F_h(b)$. The immediate reward from

transitioning is $r_h(\ell, b, 1) = \int_h^\infty v(\tau) d_h(\tau; \ell, \theta_h) d\tau - p_h(b)$ for tuple $(\ell, b, 1)$ (the winning case) and $r_h(\ell, b, \ell+1) = 0$ for tuple $(\ell, b, \ell+1)$ (the losing case). This last statement depends crucially on the independence of the incrementality across different winning rounds. The initial state of this MDP is always the lowest state, i.e., $\ell_1 = 1$.

Consequently, the optimal policy in `IB` can be computed via standard dynamic programming. Specifically, when the action set $\mathcal{B}$ has finite support (i.e., the advertiser has finite bids to choose from), the following proposition follows from the fact that MDPs with finite states, actions and horizons can be solved efficiently by dynamic programming [1].

Proposition 1. *The offline optimal policy π for any `IB` instance can be computed in poly$(|\mathcal{B}|, H)$ time by dynamic programming.*

We remark that even when $\mathcal{B} \subseteq \mathbb{R}$ is continuous with upper bound B, an ϵ-optimal policy can be computed in poly$(\frac{B}{\epsilon}, H)$ time since the reward function is continuous in $b \in \mathcal{B}$ and an ϵ-optimal action can be found at each round during the backward induction by discretizing the entire line of bid into ϵ segments. This ϵ-optimal action choices lead to an approximately optimal policy [1].

3.2 Reinforcement Learning of `IB` with Unknown Parameters

We now turn to the much more interesting and realistic situation of the online learning, particularly reinforcement learning, of the optimal `IB` solution. While we have shown that the offline version of `IB` problem can be viewed as an MDP, its online counterpart cannot be similarly tackled by standard reinforcement learning techniques due to the following novel challenges specific to the `IB` problem:

- **Challenge 1: mixed reward feedback.** Any realized learner reward, arising from a user conversion, is naturally a *mixture* effect of all the previous winning rounds, as modeled by the accumulated incrementality rate in Equation (2). This is in contrast to standard MDP, in which the reward of this round is realized independently, conditioned on its state. It is not difficult to see that the loglikelihood of our mixed rewards model is non-convex, which renders the standard RL techniques based on maximum likelihood estimation for reward parameter estimation not applicable.
- **Challenge 2: delayed reward realizations.** Moreover, the conversions in the `IB` problem follow a intricate Poisson process with heterogeneous rate and thus has delays. That is, showing an ad may not immediately lead to conversions. Such lack of immediate reward feedback renders standard Q-learning style of algorithms not easily applicable to our setup since we cannot use immediate reward feedback to update Q-functions any more.

To overcome the first challenges above, we design a novel parameter estimation methods based on moment matching to estimate θ_h's from mixed reward feedback, and prove its convergence rate in Section 4. To address the second challenge above, we design a model-based no-regret RL algorithm that integrate the parameter estimation above, and analyze its regret in Section 5.

Learner's Observations. We now formalize our learning setup. We consider the episodic online learning setting, where the learner does not know the `IB` parameters $\theta_h = \{\beta_h(\ell)\}_{\ell=1}^{h-1} \cup \{\lambda_h\}$ for each $h \in [H]$ neither the distribution of the highest *other* bid (HOB) F_h. Following the typical practice, we assume that the learner can always observe the realized HOB at any round h in each episode, denoted as m_h^t, regardless whether the learner wins or loses. This is usually termed as "*full information feedback*" in the literature of *learning to bid*.[4] Moreover, the learner also knows the set of its winning rounds denoted by an *ordered* set $\mathcal{W}^t \subset [H]$ as well as the set of the time stamps of all conversions $\mathcal{C}^t$ (a subset of $\mathbb{R}$) from each episode.

In the online learning setting, the learner aims to design a bidding algorithm $\mathcal{A}$ to minimize the expected regret defined in the following,

$$\text{Regret}(T) = T \cdot \text{OPT}(\theta, F) - \mathop{\mathbb{E}}_{(\pi^1, \pi^2, \cdots, \pi^T) \sim \mathcal{A}} \left[\sum_{t=1}^{T} R(\pi^t; \theta, F) \right], \tag{5}$$

[4]This assumption is practical. First, in online ad systems, the manual bidder can always get this feedback (also known as "minimum bid to win") no matter she wins or not, e.g., in Google Ad exchange platform [7]. Second, for the auto bidding algorithms, they are designed by the platform and it does know the realized bids at each round.

where $\mathrm{OPT}(\theta, F)$ represents the optimal expected utility achieved by the learner at each episode if the ground-truth parameters θ and F are given (recall that the MDP always starts from state $\ell = 1$), $\mathbb{E}$ represents the expectation over the randomness of algorithm $\mathcal{A}$, and π^t is the policy in the tth episode generated by the algorithm $\mathcal{A}$. Here, we slightly abuse the notation to denote $R(\pi; \widehat{\theta}, \widehat{F})$ as the expected utility achieved by policy π for estimated parameters $\widehat{\theta}$ and $\widehat{F}$.

$$R(\pi; \widehat{\theta}, \widehat{F}) = \mathop{\mathbb{E}}_{(\ell_1, \cdots, \ell_H) \sim \widehat{P}^\pi} \left[\sum_{h=1}^{H} \left(\widehat{\beta}_h(\ell_h) \cdot v - \widehat{p}_h(\pi_h(\ell_h)) \right) \cdot \widehat{F}_h(\pi_h(\ell_h)) \right], \tag{6}$$

where $\widehat{P}^\pi$ represents the joint distribution of states $(\ell_1, \cdots, \ell_H)$ induced by a bidding policy π in the MDP formulated in Section 3.1 with estimated parameters $\widehat{\theta}$ and $\widehat{F}$. Note, for second price auctions, $\widehat{p}_h(b) = b - \frac{1}{\widehat{F}_h(b)} \int_0^b \widehat{F}_h(v) dv$ (replacing F by $\widehat{F}$ in Eq. (4)). In other words, $R(\pi; \widehat{\theta}, \widehat{F})$ captures the expected total reward in the MDP when the reward function and transition probability are parameterized by $\widehat{\theta}$ and $\widehat{F}$.

4 Incrementality Parameter Estimation via Pairwise Moment-Matching

Though maximum likelihood estimation (MLE) is a natural and straightforward parameter estimation strategy, it comes with many drawbacks we want to avoid. First, the MLE criterion under our model of mixed rewards is not a convex problem and optimizing the likelihood would suffer from multiple local optimal or saddle point issues. The theoretical claims may not match the true algorithm performance. Secondly, the computational efficiency of the MLE problem is worse than what we will introduce next, which makes it less attractive in large-scale problems. Our algorithm is carefully tailored for estimating parameters of our model, given the observed trajectories of first t rounds. Specifically, the trajectory of each episode $t \in [T]$ can be described by two sequences: (1) the time stamps $\mathcal{C}^t$ of realized conversions within time $[0, \infty]$; and (2) the *ordered* subset $\mathcal{W}^t = \{w_1^t, \cdots, w_{k_t}^t\} \subseteq [H]$ which contains all the winning steps within this episode (for convenience, we extend the above notations and by letting $w_0^t = 0, w_{n_t+1}^t = H + 1$). Notably, both $\mathcal{C}^t$ and $\mathcal{W}^t$ may be empty if there is no conversion nor winning. Our estimation is based on an online algorithm: from a high-level perspective, it progresses by time intervals $[h, h+1)$ sequentially to produce estimates of parameters θ_h, by pairwisely coupling episodes and matching statistical moments. Therefore, we call it the *pairwise moment-matching* (`PAMM`) algorithm, described in Algorithm 1.

The intuition and derivation of the algorithm are given in Appendix B.In practice, terms in (8) and (9) may result in $\pm\infty$, but such an event only happens with probability tending to zero (which is why Theorem 1 has high probability guarantees). Standard numerical protection can be introduced in practice which will not change our theoretical claims. Moreover, from the implementation perspective, one start from the first interval $[0, 1)$ and move forward by running `PAMM` for each $[h, h+1)$. In each new interval, the matching step can be directly carried over by further refining the matching of the previous interval. As a result, the whole estimation procedure only involves one sweep of the data without the commuting back and forth between different time intervals. We have introduced the `PAMM` algorithm in the current form as a self-contained algorithm for estimating the inhomogeneous Poisson process, which can be of independent interest. However, the steps of `PAMM` can be easily embedded in our online learning algorithm presented in Section 5 without losing its validity.

Theorem 1 (Estimation Errors with Finite Samples)**.** *Suppose there exist positive constants C_λ and c_β, such that $\lambda_h \geq C_\lambda$ and $\beta_h(\ell) \geq c_\beta$ for any h and ℓ. Also assume that the expected number of total conversions for each episode within each interval $[h, h+1)$ is always bounded above by a constant C_T. Let $\widehat{\lambda}_h$ and $\widehat{\beta}_h$ be the estimators obtained from Algorithm 1. Then, for any $0 \leq h < H$ and any $0 \leq \ell < h$, we have the following guarantees for `PAMM`:*

1. *For any $4e^{-C_0 n_h} < \delta < 1/2$, with probability at least $1 - \delta$,*

$$|\widehat{\lambda}_h - \lambda_h| \leq C_1 \sqrt{\frac{\log 1/\delta}{n_h}}$$

2. *For any $4e^{-C_0 n_h(\ell)} < \delta < 1/2$, with probability at least $1 - \delta$,*

$$|\widehat{\beta}_h(\ell) - \beta_h(\ell)| \leq C_2 \sqrt{\frac{\log 1/\delta}{n_{h\ell}}}.$$

Algorithm 1 Pairwise Moment-Matching (PAMM) Algorithm

1: **Input**: A sample of episodes $(\mathcal{C}^t, \mathcal{W}^t), t \in [T]$ up to time $[0, h+1)$.
2: **Output:** the estimates of parameter θ_h, denoted by $\{\widehat{\beta}_h(\ell), \forall \ell \in [h]; \widehat{\lambda}_h\}$.
3: Define the set $\widetilde{\Omega}_h \subset [T]$ to be $\widetilde{\Omega}_h = \{t \in [T] : w_s^t = h \text{ for some } s \in [k_t]\}$. Set $\Omega_h = \emptyset$ and $\Lambda_h = [T] \cap \widetilde{\Omega}_h^c$.
4: **for** $t \in \widetilde{\Omega}_h$ where $w_s^t = h$ for some $s \in [H]$, apply the refining and matching procedure **do**
5: Find another episode $\phi(t) \in \Lambda_h$ such that

$$w_r^t = w_r^{\phi(t)}, 0 \le r \le s-1, w_s^{\phi(t)} > h.$$

If such a $\phi(t)$ is successfully found, remove $\phi(t)$ from Λ_h, add t to Ω_h.
6: **end for**
7: For any t, define N_h^t and $\overline{N}_h^t$ to be the number of conversions of episode t within $[h, h+\frac{1}{2})$ and $[h+\frac{1}{2}, h+1)$, respectively. For each $t \in \Omega_h$, define the two statistics

$$X_h^t = N_h^t - N_h^{\phi(t)} \quad \text{and} \quad Y_h^t = \overline{N}_h^t - \overline{N}_h^{\phi(t)}. \tag{7}$$

8: Define $\widehat{\mu}_h = \frac{1}{n_h}\sum_{t\in\Omega_h} X_h^t$ and $\widehat{\eta}_h = \frac{1}{n_h}\sum_{t\in\Omega_h} Y_h^t$. Estimate λ_h by

$$\widehat{\lambda}_h = 2(\log \widehat{\mu}_h - \log \widehat{\eta}_h). \tag{8}$$

9: **for** $0 \le \ell \le h$ **do**
10: Define the ℓth slice of Ω_h as $\Omega_{h,\ell} = \{t : t \in \Omega_h, w_{s^t-1}^t = h-\ell\}$. Set $n_h(\ell) = |\Omega_{h,\ell}|$. For each $t \in \Omega_{h,\ell}$, define $\widehat{\mu}_{h\ell} = \frac{1}{n_h(\ell)}\sum_{t\in\Omega_{h,\ell}} X_h^t$ and $\widehat{\eta}_{h\ell} = \frac{1}{n_h(\ell)}\sum_{t\in\Omega_{h,\ell}} Y_h^t$. Estimate $\beta_h(\ell)$ by

$$\widehat{\beta}_h(\ell) = \frac{\widehat{\mu}_{h\ell}^2}{\widehat{\mu}_{h\ell} - \widehat{\eta}_{h\ell}}. \tag{9}$$

11: **end for**

In the above results, C_0, C_1, C_2 are positive constants that only depend on c_β, c_λ and C_T are explicitly given in the proof.

5 The Full RL Algorithm and its Regret Analysis

We propose a reinforcement learning algorithm by incorporating our parameter estimation method PAMM and the optimal offline planning for the MDP mentioned in Section 3 to learn parameters θ, F as well as the conversion incrementality and decide the bid at each round.

Our algorithm adopts a UCB-style procedure to handle the exploration-exploitation tradeoff in incrementality bidding, which is summarized in Algorithm 2. The algorithm adopts pure explorations in several beginning episodes so that we can get enough ad impressions for each state in every round. These pure explorations are achieved by setting bids appropriately such that we can enforce to explore one (h, ℓ) pair in each exploration (Line 4 in Algorithm 2)[5]. In particular, we need $O(H^2 \log(T/\delta))$ explorations to make sure $n_h(\ell) \ge \frac{\log(4T/\delta)}{C_0}$ so that we can provide a regret bound with high probability, where C_0 is a positive constant explicitly defined in Theorem 1. Recall $n_h(\ell)$ is the total number of winnings at round h given the state is ℓ.

After pure explorations, at each episode t, we use the PAMM algorithm to get an estimator $\widehat{\theta}^t$ (resp. $\widehat{\beta}^t$) given the observed ad impression time $\mathcal{W}^t$ and the conversion time $\mathcal{C}^t$ up to episode t. In addition, we can estimate F easily using empirical distribution function and this implies we can get a uniform convergence for the transition probabilities for any action $b \in \mathcal{B}$ given any state ℓ. Then we

[5]When $\ell = h$, setting $b_0^t = 1$ means a "fake" winning at round 0 and it is consistent with our assumption that initial state $\ell_1 = 1$.

apply UCB-style RL algorithm with estimated parameters $\widehat{\theta}^t$ to decide the policy in next episode. Specifically, given the estimated $\widehat{\beta}^t, \forall h \in [H]$, we construct a confidence region centered at $\widehat{\beta}^t$ and we do offline planning for the *optimism* in the confidence region to get the policy in next episode.

Algorithm 2 Online Bidding algorithm

Input: Given δ, H, T, and C_0, C_2 defined in Theorem 1. Set $n_h(\ell) = 0, \forall h \in [H], \ell \in [h]$.
for Episode $t = 1, \cdots, T$ **do**
  if $\exists h, \ell, n_h(\ell) < \frac{\log(4T/\delta)}{C_0}$ **then**
    Set bids, s.t. $b_h^t = b_{h-\ell}^t = 1$, and $b_{h'}^t = 0, \forall h' \neq h, h-\ell$. `[Pure Exploration]`
  else
    For $h \in [H]$, observe state ℓ_h^t and set the bid $b_h^t = \pi_h^t(\ell_h^t)$.
  end if
  Observe the set of the ad impression time $\mathcal{W}^t$, the set of the conversion time $\mathcal{C}^t$, and the vector of HOBs $m^t = \{m_h^t, \forall h \in [H]\}$.
  Update $n_h(\ell)$ given $\mathcal{W}^t$.
  Update $\widehat{\theta}^t = \{\widehat{\beta}_h^t(\ell), \forall \ell \in [h]; \widehat{\lambda}_h^t\}$ through `PAMM` method (Algorithm 1) given $(\mathcal{W}^t, \mathcal{C}^t)$.
  Update $\widehat{F}_h^t(b) = \frac{1}{t}\sum_{s=1}^t \mathbb{I}\{m_h^s \leq b\}, \forall b$.
  if $\forall h \in [H], \ell \in [h], n_h(\ell) \geq 1$ **then**
    Construct confidence region $\widehat{\mathrm{CR}}^t$ for parameters θ, s.t.

$$\widehat{\mathrm{CR}}^t = \left\{ (\widehat{\beta}, \widehat{\lambda}) : \forall h, |\widehat{\beta}_h(\ell) - \widehat{\beta}_h^t(\ell)| \leq C_2 \sqrt{\frac{\log(T/\delta)}{n_h(\ell)}} \right\}$$

    Compute the bidding policy π^{t+1} for $(t+1)$th episode, s.t.,

$$\pi^{t+1} = \operatorname*{argmax}_{\pi} \max_{\widehat{\theta} \in \widehat{\mathrm{CR}}^t} R(\pi; \widehat{\theta}, \widehat{F}^t) \tag{10}$$

  end if
end for

Remark 2 (Computational Efficiency of Alg. 2)**.** *An important observation is that $R(\pi; \widehat{\theta}, \widehat{F})$ is non-decreasing in $\widehat{\beta}_h(\ell)$ (see Eq. (6)) for any fixed π and $\widehat{F}$. Therefore, the optimal choice of $\widehat{\beta}$ in Eq. (10) in Algorithm 2 is to set $\widehat{\beta}_h(\ell) = \widehat{\beta}_h^t(\ell) + C_2\sqrt{\frac{\log(1/\delta)}{n_h^t(\ell)}}$, and π^{t+1} will be the optimal policy for the MDP parameterized by $\widehat{F}^t$ and $\widehat{\beta}$, and thus can be computed efficiently (Proposition 1).*

Regret Analysis. Combining the convergence rate analysis from Theorem 1 and a novel regret decomposition technique (necessary for our particular setup), we are able to prove the regret guarantee of our online bidding algorithm. Our regret bound is summarized in Theorem 2, the proof of which is deferred to Appendix C.

Theorem 2. *Under Assumption 1, for any fixed δ, we have with probability at least $1-\delta$, the regret achieved by our Algorithm 2 is bounded by*

$$\mathrm{Regret}(T) \leq O(H^2\sqrt{\log(HT/\delta)T} + H^2\log(T/\delta)).$$

6 Discussions and Future Work

Our RL algorithm lies in the model-based RL algorithm literature. The best known regret bound for model-based RL is $\widetilde{O}(\sqrt{H^2SAT})$ [2], where S is the number of states and A is the number of actions. Since $S = H$ in our case, our regret bound is slightly worse from the best possible by an factor of $\sqrt{H}$, but is independent of the number of actions. This may be due to the complexity of our model. Whether we can close this gap is an interesting open question. As we mentioned earlier, our algorithm can be extended to handle continuous bidding space and our regret analysis still works. In this paper, we assume the conversion rate function d_h doesn't depend on the shown ads of other bidders but only on the learner's *last* ad impression. Relaxing these assumptions may result in very interesting future work.

References

[1] A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun. Reinforcement learning: Theory and algorithms. *Working Draft*, 2019.

[2] M. G. Azar, I. Osband, and R. Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 263–272. PMLR, 06–11 Aug 2017.

[3] A. Badanidiyuru, A. Evdokimov, V. Krishnan, P. Li, W. Vonnegut, and J. Wang. Handling many conversions per click in modeling delayed feedback. *AdKDD*, 2021.

[4] A. Badanidiyuru, Z. Feng, and G. Guruganesh. Learning to bid in contextual first price auctions. *CoRR*, abs/2109.03173, 2021.

[5] S. Balseiro, N. Golrezaei, M. Mahdian, V. Mirrokni, and J. Schneider. Contextual bandits with cross-learning. In *Advances in Neural Information Processing Systems 32*, pages 9679–9688. 2019.

[6] O. Chapelle. Modeling delayed feedback in display advertising. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1097–1105, 2014.

[7] G. Developers. Real-time bidding. *https://developers.google.com/authorized-buyers/rtb/request-guide*, 2022.

[8] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1): 242–259, March 2007.

[9] Facebook. About bid strategies. *https://www.facebook.com/business/help/403 1619591734742116?id=2196356200683573*, 2022.

[10] Z. Feng, C. Podimata, and V. Syrgkanis. Learning to bid without knowing your value. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, page 505–522, 2018.

[11] Google. Autobidding support page. *https://support.google.com/google-ads/405 answer/2979071*, 2022.

[12] Y. Han, Z. Zhou, A. Flores, E. Ordentlich, and T. Weissman. Learning to bid optimally and efficiently in adversarial first-price auctions. *CoRR*, abs/2007.04568, 2020.

[13] Y. Han, Z. Zhou, and T. Weissman. Optimal no-regret learning in repeated first-price auctions. *CoRR*, abs/2003.09795, 2020.

[14] B. Howson, C. Pike-Burke, and S. Filippi. Delayed feedback in episodic reinforcement learning. *arXiv preprint arXiv:2111.07615*, 2021.

[15] P. Jain and P. Kar. Non-convex optimization for machine learning. *arXiv preprint arXiv:1712.07897*, 2017.

[16] T. Lancewicki, A. Rosenberg, and Y. Mansour. Learning adversarial markov decision processes with delayed feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[17] R. A. Lewis and J. Wong. Incrementality bidding & attribution. *Available on SSRN*, 2018.

[18] T. Nedelec, C. Calauzènes, N. E. Karoui, and V. Perchet. Learning in repeated auctions. *Foundations and Trends® in Machine Learning*, 15(3):176–334, 2022. ISSN 1935-8237. doi: 10.1561/2200000077.

[19] G. Noti and V. Syrgkanis. Bid prediction in repeated auctions with learning. In *Proceedings of the Web Conference 2021*, WWW '21, page 3953–3964, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127.

[20] Y. Saito, G. Morisihta, and S. Yasui. *Dual Learning Algorithm for Delayed Conversions*, page 1849–1852. 2020.

[21] Y. Su, L. Zhang, Q. Dai, B. Zhang, J. Yan, D. Wang, Y. Bao, S. Xu, Y. He, and W. Yan. An attention-based model for conversion rate prediction with delayed feedback via post-click calibration. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021.

[22] H. R. Varian. Position auction. *International Journal of Industrial Organization*, 2006.

[23] J. Weed, V. Perchet, and P. Rigollet. Online learning in repeated auctions. In *29th Annual Conference on Learning Theory*, pages 1562–1583, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]