Real-Time Bidding in Online Advertising using Reinforcement Learning: Optimizing Bidding Strategies

Chitransh Kumar
202211015

Kartik Chugh
202211038

Kushagra Taneja
202211042

Nitesh Parihar
202211058

Repository Link For Code

***Abstract*—In the realm of digital advertising, real-time bidding (RTB) has emerged as a pivotal mechanism for optimizing ad impressions and maximizing advertiser objectives. This paper presents a framework for developing an optimal bidding strategy through reinforcement learning, focusing on enhancing user engagement, particularly through clicks, as the primary key performance indicator (KPI). The bidding process is modeled as an episodic decision-making task, wherein each episode comprises a series of sequential auctions characterized by a high-dimensional feature vector that encapsulates relevant contextual information such as user demographics, time, and campaign attributes.**

At each auction, a bidding agent must strategically decide its bid price while navigating constraints related to budget and the number of remaining auctions. The decision-making process is informed by the predicted click-through rate (pCTR), which serves as an expected reward metric. By formulating the problem within a reinforcement learning framework, the agent learns to balance the trade-offs between bidding aggressiveness and budget management to optimize user responses across multiple auctions. Our findings demonstrate that the proposed approach significantly enhances the bidding agent's ability to maximize clicks while adhering to budgetary limits, thereby contributing to the overall effectiveness of RTB campaigns in the digital advertising landscape.

***Index Terms*—Real-Time Bidding (RTB), Reinforcement Learning (RL), Q-learning, Online Advertising, Bidding Strategy**

I. Introduction

Bidding in display advertising can be viewed as a decision-making process that unfolds over multiple rounds, or episodes. Each episode consists of T sequential auctions, where the agent must carefully manage its strategy. Every auction is characterized by a high-dimensional feature vector (x), which includes critical information such as location, time, and campaign-specific details. The agent is provided with a fixed budget (B) at the start, with the primary objective of maximizing clicks across the auctions.

At each auction, the agent evaluates three key elements:

- t: The number of auctions remaining in the episode.
- b: The current unspent budget.
- x: The auction-specific feature vector.

Based on these factors, the agent determines a bid price (a). If the bid price $a \geq \delta$ (the market price), the agent wins the auction and pays δ, reducing the remaining budget. If $a < \delta$, the agent loses the auction without spending any budget. The predicted click-through rate (pCTR), represented as $\theta(x)$, provides an estimate of the expected reward for winning the auction.

The episode concludes either when all auctions have been processed or the budget is exhausted, after which the process resets for the next episode.

II. Problem Statement

Advertisers in RTB face a complex optimization problem. They must decide on a bid price for each ad impression while balancing budget constraints and the goal of maximizing a specific KPI. A bid request is represented by a feature vector x, containing information such as user attributes, time, and context. Advertisers must predict the performance of each bid (e.g., expected clicks or conversions) and select the best bidding strategy accordingly. The challenge is to dynamically optimize this process while maintaining cost efficiency.

III. MDP Formulation of RTB

A Markov Decision Process (MDP) provides a framework that is widely used for modeling agent-environment interactions. Our notations are listed in Table I. An MDP can be represented by a tuple $(S, \{A_s\}, \{P^a_{ss'}\}, \{R^a_{ss'}\})$, where S and A_s are two sets of all states and all possible actions in state $s \in S$, respectively. $P^a_{ss'}$ represents the state transition probability from state $s \in S$ to another state $s' \in S$ when taking action $a \in A_s$, which is denoted by $\mu(a, s, s')$. Similarly, $R^a_{ss'}$ is the reward function denoted by $r(a, s, s')$, representing the reward received after taking action a in state s and then transitioning to state s'.

We consider (t, b, x_t) as a state s_1 and assume the feature vector x_t is drawn i.i.d. from the probability density function $p_x(x)$. The full state space is $S = \{0, \cdots, T\} \times \{0, \cdots, B\} \times X$. If $t = 0$, the state is regarded as a terminal state, meaning the end of the episode. The set of all actions available in state (t, b, x_t) is $A(t, b, x_t) = \{0, \cdots, b\}$, corresponding to the bid price. Furthermore, in state (t, b, x_t) where $t > 0$, the agent, when bidding a, can transit to $(t-1, b-\delta, x_{t-1})$ with probability $p_x(x_{t-1})m(\delta, x_t)$ where $\delta \in \{0, \cdots, a\}$ and $x_{t-1} \in X$.

This corresponds to winning the auction and receiving a reward $\theta(x_t)$. The agent may also lose the auction and transit to $(t-1, b, x_{t-1})$ with probability $p_x(x_{t-1}) \sum_{\delta=a+1}^{\infty} m(\delta, x_t)$, where $x_{t-1} \in X$. All other transitions are impossible due to the auction process.

In summary, the transition probabilities and reward function can be written as:

$$\mu\left(a,(t,b,x_t),(t-1,b-\delta,x_{t-1})\right) = p_x(x_{t-1})m(\delta,x_t), \quad (1)$$

$$\mu\left(a,(t,b,x_t),(t-1,b,x_{t-1})\right) = p_x(x_{t-1}) \sum_{\delta=a+1}^{\infty} m(\delta,x_t), \quad (2)$$

$$r\left(a,(t,b,x_t),(t-1,b-\delta,x_{t-1})\right) = \theta(x_t), \quad (3)$$

$$r\left(a,(t,b,x_t),(t-1,b,x_{t-1})\right) = 0, \quad (4)$$

where $\delta \in \{0, \cdots, a\}$, $x_{t-1} \in X$, and $t > 0$. Specifically, the first equation describes the transition when placing a bid price $a \geq \delta$, while the second equation describes the transition when losing the auction.

A deterministic policy π is a mapping from each state $s \in S$ to an action $a \in A_s$, i.e., $a = \pi(s)$, which corresponds to the bidding strategy in RTB display advertising. According to policy π, we define the value function $V^{\pi}(s)$ as the expected sum of rewards when starting in state s and following policy π. This satisfies the Bellman equation with the discount factor $\gamma = 1$, since in our scenario the total number of clicks is the optimization target, regardless of the time of the clicks:

$$V^{\pi}(s) = \sum_{s' \in S} \mu(\pi(s), s, s') \left[r(\pi(s), s, s') + V^{\pi}(s') \right]. \quad (5)$$

The optimal value function is defined as:

$$V^*(s) = \max_{\pi} V^{\pi}(s). \quad (6)$$

We also define the optimal policy as:

$$\pi^*(s) = \arg\max_{a \in A_s} \left\{ \sum_{s' \in S} \mu(a, s, s') \left[r(a, s, s') + V^*(s') \right] \right\}, \quad (7)$$

which gives the optimal action in each state s, and $V^*(s) = V^{\pi^*}(s)$. The optimal policy $\pi^*(s)$ represents the optimal bidding strategy that we aim to find. For notation simplicity, we use $V(s)$ to denote the optimal value function $V^*(s)$ in the later sections.

TABLE I
A SUMMARY OF OUR NOTATIONS.

Notation	Description
x	The feature vector that represents a bid request.
X	The whole feature vector space.
$p_x(x)$	The probability density function of x.
$\theta(x)$	The predicted CTR (pCTR) if winning the auction for x.
$m(\delta, x)$	The probability density function of market price δ given x.
$m(\delta)$	The probability density function of market price δ.
$V(t, b, x)$	The expected total reward with starting state (t, b, x), taking the optimal policy.
$V(t, b)$	The expected total reward with starting state (t, b), taking the optimal policy.
$a(t, b, x)$	The optimal action in state (t, b, x).

Algorithm 1 Reinforcement Learning to Bid (Value Function Calculation)

Input: p.d.f. of market price $m(\delta)$, average CTR θ_{avg}, episode length T, budget B
Output: value function $V(t, b)$
Initialize $V(0, b) = 0$ {Set the value function for $t = 0$}
for $t = 1, 2, \ldots, T-1$ **do** {Iterate over each time step}
  for $b = 0, 1, \ldots, B$ **do** {Iterate over budget}

$$V(t,b) \approx \max_{0 \leq a \leq b} \left\{ \begin{array}{l} \sum_{\delta=0}^{a} m(\delta)\theta_{\text{avg}} + \sum_{\delta=0}^{a} m(\delta) V(t-1, b-\delta) \\ + \sum_{\delta=a+1}^{\infty} m(\delta) V(t-1, b) \end{array} \right\}$$

  end for
end for=0

Algorithm 2 Optimal Bid Price Calculation

Input: CTR estimator $\theta(x)$, value function $V(t, b)$, current state (t_c, b_c, x_c)
Output: optimal bid price a_c in current state
Calculate the pCTR for the current bid request: $\theta_c = \theta(x_c)$
for $\delta = 0, 1, \ldots, \min(\delta_{\max}, b_c)$ **do** {Enumerate possible bid prices}
  if $\theta_c + V(t_c - 1, b_c - \delta) - V(t_c - 1, b_c) \geq 0$ **then** {Check if the action is optimal}
    $a_c \leftarrow \delta$ {Set the optimal bid price}
  end if
end for=0

IV. Challenges and Limitations

A. Scalability Issues

One of the main challenges in real-time bidding (RTB) is the large volume of auctions and the corresponding state-action space in reinforcement learning (RL). This makes it computationally expensive and resource-intensive to compute the exact value function for each possible state-action pair. The original dynamic programming-based solution struggles to scale effectively for large-scale auctions.

B. Function Approximation Errors

To tackle the large state space, neural networks are employed for function approximation. However, these approximations may introduce errors in prediction, particularly in larger scales. As demonstrated in the experiments, when the auction volume and budget are significantly large, the neural network approximation (RLB-NN) does not perform as well due to poor generalization.

C. Dynamic Nature of the Environment

The RTB environment is highly dynamic, and the data distribution can vary over time. This makes it difficult to model the true market conditions with static assumptions, affecting the reliability of the model. The authors note that existing approaches like model-free RL methods suffer from reward sparsity and environment stochasticity.

D. Handling Budget Constraints

Budget pacing is critical in RTB, as the agent must balance between spending the budget too quickly or too slowly. Incorrect pacing can lead to underutilization of the budget or exhausting it prematurely, impacting the overall effectiveness of the bidding strategy.

E. Feature Dependency

The effectiveness of the bidding policy heavily depends on accurately predicting the click-through rate (CTR) and market price distributions. Inaccurate estimations of these features can degrade the bidding performance.

V. Conclusion

The paper proposes a model-based reinforcement learning (RLB) approach to optimize bidding strategies in real-time bidding for display advertising. By formulating the problem as a Markov Decision Process (MDP), the bidding strategy is treated as a sequential decision-making process where the optimal bid is derived using dynamic programming. The authors address the scalability issue by leveraging neural network approximations for the value function and introducing coarse-to-fine episode segmentation and state mapping models to handle large-scale data effectively.

The experimental results on both small- and large-scale datasets show that the proposed RLB approach outperforms traditional methods such as linear bidding strategies and model-free methods. However, limitations remain in terms of scalability and the accuracy of neural network approximations in larger environments. For future work, the authors propose investigating model-free approaches like Q-learning and policy gradient methods to unify utility estimation and bidding optimization within a single framework. Additionally, the reinforcement learning model can naturally handle budget pacing, a critical challenge in RTB, which makes it a promising approach for optimizing click performance in dynamic environments.

VI. References

References

[1] **Real-Time Bidding by Reinforcement Learning in Display Advertising** (2016).

[2] **Deep Reinforcement Learning for Search, Recommendation, and Online Advertising: A Survey** (2020).

[3] **Optimal Real-Time Bidding for Display Advertising** (2017).