Real-Time Bidding in Online Advertising using RL

Optimizing Bidding Strategies with Reinforcement Learning

Group: Agent

Indian Institute of Information Technology Vadodara International Campus Diu

October 26, 2024

Group Information

Group Name: Agent
Team Members:

- Chitransh Kumar (202211015)
- Kartik Chugh (202211038)
- Kushagra Taneja (202211042)
- Nitesh Parihar (202211058)

Outline

Problem Statement

- **Overview:** In real-time display advertising, advertisers compete in auctions to display ads to targeted users within a set budget. Each bid request is represented by a feature vector containing details such as user, location, and time. Advertisers predict a Key Performance Indicator (KPI) like Click-Through Rate (CTR) or Conversion Rate (CVR) using historical data to optimize bids. The challenge is to develop an optimized bidding strategy that maximizes the chosen KPI while staying within the campaign's budget and targeting constraints.
- **Goal:** To develop a smart bidding agent that optimizes bids based on real-time data, maximizing campaign performance while managing budget constraints.

Research Papers

- **Real-Time Bidding by Reinforcement Learning in Display Advertising** (2016)
- **Deep Reinforcement Learning for Search, Recommendation, and Online Advertising: A Survey** (2020)
- **Optimal Real-Time Bidding for Display Advertising** (2017)

Input, Output, and End Goal

Input:

- Auction Data: Historical data on ad impressions, clicks, conversions, and costs.
- User Context: Information about the user, including demographics and browsing behavior.
- Campaign Parameters: Current budget, remaining campaign duration, and performance metrics.
- Time and Location

Output:

- Optimal Bid Prices: Recommended bid amounts for each ad impression.

End Goal: To develop a RL-based bidding agent that optimally determines bid prices for online ad impressions in real-time, maximizing ROI and adapting to dynamic market conditions.

Tech Stack

- **Programming Languages:** Python
- **Libraries/Frameworks:** PyMDPToolbox, NumPy

Key Takeaways from Paper-1

- **Real-Time Auctions:** Ads are auctioned instantly, requiring smart bidding strategies.
- **Reinforcement Learning (RL):** Treats bidding as a learning problem to optimize bid prices based on feedback from ad auctions.
- **Strategic Bidding Factors:**
  - Ad value (CTR)
  - Remaining budget
  - Future opportunities
- **Challenges of Simple Strategies:**
  - Competition and auction volume variability
  - Limitations of static mathematical models

Key Takeaways from Survey Paper

- **DRL in Advertising:** Leveraging Deep Reinforcement Learning (DRL) enhances real-time information-seeking techniques.
- **Continuous Strategy Updates:** DRL adapts strategies based on user feedback, optimizing for metrics like click-through rates and user engagement.
- **Core RL Concepts:**
  - **Multi-Armed Bandits (MAB):** Models exploration/exploitation in simple environments.
  - **Markov Decision Processes (MDP):** Structures sequential decision-making with state transitions.

Input Parameters

- Budget (B): Total budget for the ad campaign.
- Episode Length (T): Number of auctions or bid requests in an episode.
- Feature Vector (x): Represents user and ad context (e.g., location, time).
- Market Price (m): Price distribution based on market conditions.

Notation

- 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).

MDP Formulation of RTB

- States (S): Defined by remaining auctions, budget, and user features.
- Actions (A): Set bid price for each auction based on current state.
- Rewards (R): Estimated CTR if the auction is won.
- Transitions (P): Determine changes in state and budget after each bid.

Working - Bidding Strategy

- Use Markov Decision Process (MDP) to learn optimal bidding policy.
- Policy selects bid prices to balance budget constraints and maximize rewards.
- Dynamic programming and neural network approximations handle scalability.

Value Function Formulation

- Transition probabilities:

$$\mu(a,(t,b,x),(t-1,b-\delta,x')) = p_x(x')m(\delta,x)$$

- Reward function:

$$r(a,(t,b,x),(t-1,b-\delta,x')) = \theta(x)$$

- Bellman equation:

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

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 $V(t,b)$

Algorithm 1 Value Function Calculation

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

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

  end for
end for

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

Algorithm 2 Optimal Bid Price Calculation

1: **Input:** CTR estimator $\theta(x)$, value function $V(t,b)$, current state (t_c, b_c, x_c)
2: **Output:** optimal bid price a_c in current state
3: Calculate the pCTR for the current bid request: $\theta_c = \theta(x_c)$
4: **for** $\delta = 0, 1, \ldots, \min(\delta_{\max}, b_c)$ **do** ▷ Enumerate possible bid prices
5: **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
6: $a_c \leftarrow \delta$ ▷ Set the optimal bid price
7: **end if**
8: **end for**

Conclusions

- RL-based bidding strategies outperform traditional methods in RTB.
- The model dynamically adjusts bids based on auction and budget constraints.
- Future Work: Exploring model-free approaches for more flexible solutions.