Improving Real-Time Bidding Using a Constrained Markov Decision Process

Manxing Du[1(✉)], Redouane Sassioui[1], Georgios Varisteas[1], Radu State[1], Mats Brorsson[2], and Omar Cherkaoui[3]

[1] University of Luxembourg, Luxembourg City, Luxembourg
{manxing.du,redouane.sassioui,georgios.varisteas,radu.state}@uni.lu
[2] Royal Institute of Technology (KTH), Stockholm, Sweden
matsbror@kth.se
[3] University of Quebec in Montreal, Montreal, Canada
cherkaoui.omar@uqam.ca

Abstract. Online advertising is increasingly switching to real-time bidding on advertisement inventory, in which the ad slots are sold through real-time auctions upon users visiting websites or using mobile apps. To compete with unknown bidders in such a highly stochastic environment, each bidder is required to estimate the value of each impression and to set a competitive bid price. Previous bidding algorithms have done so without considering the constraint of budget limits, which we address in this paper. We model the bidding process as a Constrained Markov Decision Process based reinforcement learning framework. Our model uses the predicted click-through-rate as the state, bid price as the action, and ad clicks as the reward. We propose a bidding function, which outperforms the state-of-the-art bidding functions in terms of the number of clicks when the budget limit is low. We further simulate different bidding functions competing in the same environment and report the performances of the bidding strategies when required to adapt to a dynamic environment.

Keywords: Display Advertising · Real-time bidding · Markov Decision Process · Reinforcement Learning

1 Introduction

The share of digital ad spending in the global ad market has increased tremendously in the recent years and is expected to soar up to over 46% by 2020 as eMarketer forecasts [16]. Programmatic platforms like *Real-time bidding* (RTB) gradually takes over as the major tool for the digital ad trading [13]. Instead of bidding on keywords like in sponsored search [3], or on the context of the website as in contextual advertising [8], RTB targets the best match of users and campaigns at each ad impression level.

In an RTB system, the *demand-side platform* (DSP) plays the role of bidding for ad impressions on behalf of the advertisers. An ad exchange (ADX) receives bids

M. Brorsson—Work done while Mats Brorsson was at OLAmobile, Luxembourg.

G. Cong et al. (Eds.): ADMA 2017, LNAI 10604, pp. 711–726, 2017.
https://doi.org/10.1007/978-3-319-69179-4_50

from DSPs and holds second-price auctions; the DSP with the highest bid wins the auction but pays the second highest price, known as the *market price*. According to the Bayesian-Nash equilibrium in the auction theory [14], each bidder's optimal strategy in a second-price auction is to bid the value of each impression evaluated from its own perspective. This is known as truth-telling bidding.

However, in reality, truth-telling bidding may not be the optimal solution due to the budget limit for each ad campaign. Bidding constantly at the true value can lead to running out of budget quickly without having covered a wide range of users and impressions [20]. Consequently, the bidder fails to obtain potential profits and might even be subject to heavy losses since the payback of the impressions may be less than the total cost of winning the auction.

The optimization of bidding strategies has been widely studied in the computational advertising industry [9,21]. The goal of an optimal bidding strategy is to intelligently set the bid price for each ad auction in order to maximize the total number of clicks or profits [17] within a certain budget. This optimization problem fits perfectly into the framework of a *Constrained Markov Decision Process* (CMDP) [2], which allows to maximize one criterion while keeping another criterion below a given threshold.

In this paper, we cast the optimization of the sequential bid requests as a CMDP. This is done in order to find the optimal bid price under budget constraints for each auction. A CMDP is defined by the tuple $< S, A, P, R, C, V >$, which correspondingly represents state, action, state transition probability, reward, cost, and the value of the constrains. We consider the predicted *click-through-rate* (CTR[1]) as the *state*, the *number of clicks* as the *reward* to maximize, the *market price* as the *cost*, and the budget limit as the *constraint*. We integrate the optimization problem and the condition of budget limit into the model and use the linear programming method [11] to solve the CMDP. The policy derived from the solution gives an optimal bid price for each state.

Our contributions are summarized as follows:

- We formalize the bidding optimization problem as a CMDP which optimizes the bidding performance on the impression level. Instead of directly using the features from the impression space, our approach simplifies and limits the state as the discretized predicted CTR. This results in a significant decrease of the dimensionality of the state space. Another outcome is that we maximize the number of clicks within the constrained budget.
- We introduce the use of conditional market price distribution derived from the joint distribution of historical market price and the predicted CTR. This captures the correlation between the winning probability and the user level information.
- We show how the well-tuned bidding functions handle the dynamics of the market price, by simulating scenarios where different bidders compete with each other in the same environment. Previous studies compare the bid price of their proposed bidding strategies with only the historical winning price.

[1] The CTR can be seen as the probability of a user clicking on the ad being shown. The predicted CTR is a prediction of this probability based on features of the publisher site/app and the user visiting it.

2 Related Work

A bidding strategy is one of the key components of online advertising [3,12,21]. An optimal bidding strategy helps advertisers to target the valuable users and to set a competitive bid price in the ad auction for winning the ad impression and displaying their ads to the users. If the ads are clicked by the users or the users make purchases after clicking the ads, profits will be generated for the bidder.

The linear bidding strategy is widely used with real-world applications [17,21]. This strategy bids proportionally higher for bid requests with higher estimated CTR, failing however to deal with the budget constraints. In [21], a non-linear bidding function is proposed to adapt different budget constraints. It is shown to outperform the linear bidding function in terms of the number of clicks per ad campaign, however the winning probability function does not describe well the real winning price distribution. We are addressing this shortcoming with CMDP, since we derive the winning price distribution directly from the historical data and use it in the bid optimization process.

Reinforcement learning methods have been widely applied on solving decision making problems in online advertising applications. The models fall into two major frameworks, namely the Multi-Armed Bandits (MAB) [19] and the *Markov Decision Process* (MDP). In both models, the key components are the states, actions, and rewards. Several prior works have tried to formalize the online advertising problem as a reinforcement learning framework. In [11,18], the authors fit the banner delivery and the ad allocation problems into the MAB model while the rewards are the number of ad clicks and the profits. However, these prior works, assume no cost for showing the impressions and thus consider no constraints. This cost is highly important for an RTB system. Additional user-level information is also neglected in the previous works, which are of paramount importance for pricing the value of an ad impression, so that profitable customers are targeted.

In sponsored search, ad impressions are shown with certain costs, namely the market price, and the ad agent bids on keywords. The ad agent first needs to place a bid to win the auction, such that its ads can be displayed to the users. In [4], the authors proposed a bidding function for sequential bidding requests in sponsored search. That paper addressed the problem of right-censorship for the market price in the second-price auction scenario. The market price is right-censored because only the winner of the auction is informed; for lost auctions, the bidders only know that the market price is equal or higher than their own bid price. The authors formalized keyword bidding as a MDP, where the number of auctions and the budget limit are the states, the discretized bid price are the actions, and the total number of clicks are the rewards. With RTB, auctions are held for each single impression, thus the budget per auction needs to be constrained to optimize spending relative to profit. In our work, the CMDP model extends MDP by accounting for budget constraints directly and implicitly takes the impression level information in the predicted CTR to find the optimal bid price.

The CTR estimation reflects how well the user and the publisher match the targeting goal of each campaign. It directly impacts the bid price, which subsequently affects the winning price. The authors in [7] formulated a reinforcement learning based bidding function, by extending the concept in [4] and applying it into the RTB system. However, they implicitly correlate the user features to the winning rate approximation by multiplying the average CTR to the density function of the market price. In our work, we discuss the importance of correlating market price with the CTR and directly take the discretized CTR as the state for the bidding optimization. In addition, their bid price is set in two steps: state value lookup and action calculation in [7]. In contrast, our model solved the bidding optimization problem with linear programming which derives the optimal bid price for each state; thus the bid price can be set after a single lookup per bid request.

3 Background and Bidding Strategy Modeling

In an RTB system, whenever a user visits the publisher's website, the ad slots on the website are sold through real-time auctions. A bid request is sent from user's browser to an ad exchange, which contains the user profile, the publisher side information, and the description of the ad slot. The ADX distributes the bid requests to multiple DSPs which bid on behalf of advertisers. Each DSP derives the high dimensional feature vectors from the bid request and estimates the probability that the user clicks (or purchases) the ad campaign selected for bidding. The DSP integrates the CTR prediction, the budget, and the winning probability estimation to compute a bid price sending back to the ADX. An ADX holds a second-price auction, which selects the winner of the auction as the DSP with the highest bid and sends the URL of the corresponding ad back to the publisher. In a second-price auction, the second highest bid is denoted as the market price or the winning price. In this paper, we use market price and winning price interchangeably. The ad from the winner will be shown to the user. If the user clicks the ad, he/she will be redirected to the landing page of the ad. On the landing page, the user may complete subscription or purchase depending on the property of the ad. User's activity of ad clicks or purchase will be sent back to the DSP as feedback. DSPs correspondingly use the feedback to adjust their bidding strategies. Figure 1 shows these interactions graphically.

The interaction between the bidder and the ADX can be framed as an interaction between an agent and an environment, similar to the reinforcement learning framework [19]. As shown in Fig. 1, an agent (a DSP) receives a state (bid request) from the environment and takes an action (sets a bid price) which triggers the environment to respond with a reward (feedback per auction) and the next state (next bid request). The goal of the DSP is to learn an optimal mapping from a bid request to a bid price, which maximizes the reward it receives over a finite time period (an episodic task) or an infinite time period (a continuous task). In our work, the end of the time period is when the budget is exhausted.

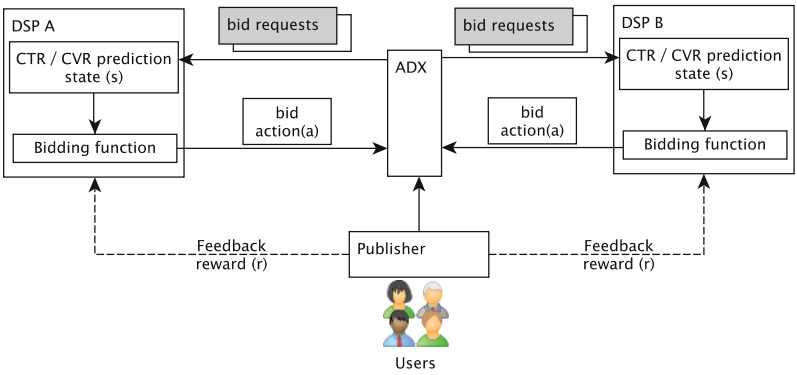

Fig. 1. A RTB system overview

3.1 Learning with Constraints

When no additional constraints exist, reinforcement learning is usually formalized as a Markov decision process (MDP) [19]. However, RTB requires to keep the budget under certain constraints and in the meantime maximize the total number of clicks. A Constrained Markov Decision Process (CMDP) is a class of MDP models which can set more than one conflicting objectives. A typical case of CMDP is the situation where we want to maximize one criterion while keeping another below a given threshold. Therefore we relied on such models to describe the bidding function.

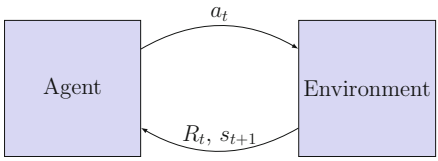

Fig. 2. Graphical representation of an MDP. At each time t the agent knows the environment state s_t and based on the transition probability model, it takes action a_t and receives the reward R_t and observe the next state s_{t+1}.

Figure 2 shows a graphical representation of a MDP at time t. A CMDP is defined by the tuple $< S, A, P, R, C, V >$.

- S is the state set.
- A is the action set.
- $P(s'|s, a)$ is the transition probability function, such that $P(s'|s, a)$ is the probability that the system moves to state s' given that it is in state s and the agent takes an action a.

- $R(s, a)$ is the expected reward to maximize, when the system is in state s and action a is taken.
- C is the constraint cost function. $C(s, a)$ is the expected cost acquired when the system is in state s and the agent chooses an action a.
- V is a vector of values that correspond to each constraint.

A *policy* is defined as a function $\pi : S \mapsto A$ which maps the state space S to the action space A, and specifies the action $a = \pi(s)$ that the agent will choose when being in state s.

The objective of a CMDP is to solve the following optimization problem

$$\begin{aligned} \max_{\rho} \quad & \overline{R} = \sum_{s,a} \rho(s,a) R(s,a) \\ \text{s.t} \quad & \sum_{s,a} \rho(s,a) C(s,a) \leq V \\ \text{and} \quad & \sum_{s \in S} \sum_{a \in A(s)} \rho(s,a) = 1 \end{aligned} \tag{1}$$

where ρ is a vector of length $|S| * |A|$ in which each element corresponds to the probability of being in state s and taking action a. Let $\tilde{\rho}$ be the optimal solution of Eq.(1), the optimal policy $\tilde{\pi}$ to apply in each state is the action a which is in the $\tilde{\rho}(s, a)$.

3.2 RTB as a Model-Based CMDP

As described above, an optimal bidding function combines the CTR estimation, the winning probability, and the budget constraint to set a bid price for each bid request. The dynamics of the RTB system depend on the diversity of the users and the behavior of all other bidders. The user diversity is reflected in the high dimensional feature vector derived from each bid request. The market price, which is the highest among all loosing bids, determines how much the winner pays for the winning auction, in other words, how much budget is spent. The historical market price distribution can be used to estimate the winning probability of bidding a certain price.

Directly using the high dimensional user feature vector $\mathbf{x}$ as the state in the Markov model is very difficult because of the sparsity of the data. However, mapping this feature vector into a lower dimensional space is possible through the CTR prediction $\theta(x)$. The latter takes the feature vector $\mathbf{x}$ as input and calculates the probability of a click. This method has been used to optimize a non-linear bidding strategy [21]. The underlying assumption is that the state dynamics of the RTB system can be completely captured by CTR. Obviously, both the bidding strategy and the winning rate estimation are dependent on the CTR.

We therefore also assume that user dynamics are described by the predicted CTR $\theta(x)$ and thus project the high dimensional feature space into an 1-dimensional space. The predicted CTR is the state of the RTB system, $S = \Theta$. The set of actions, A, consists of the set of permitted bids, i.e.,

$A = \{0, 1, 2, ..., a_{max}\}$, where a_{max} is the maximum bid that a bidder wants to pay for showing its ad. The transition probability function P equals the probability density function (pdf) of θ and is independent of the current state and the action taken. Formally:

$$P(\theta'|\theta, a) = p_{CTR}(\theta') \tag{2}$$

where p_{CTR} is the pdf of the predicted CTR. p_{CTR} can be approximated from historical data using a kernel density estimation.

The reward of an RTB system is usually defined by the advertisers. For branding purpose, the goal of the advertisers can be to maximize the number of ad impressions. However, more commonly, the advertisers are not satisfied by only displaying their ads. Thus, they set the goal as acquiring user interactions like clicks or even further, purchases. In this paper, the reward of an RTB system is the number of clicks. Since for example, in the iPinYou dataset, there are 5 out of 9 campaigns without any purchase in both training and test datasets. It is a chain process to calculate the expected reward. Firstly, the bidder needs to win an ad auction by placing a bid. The winning probability is derived from the market price distribution. After winning the auction, the expected reward is given by the estimated CTR. The cost is defined as the market price each bidder pays for a winning auction. If the bid price of a bidder is not the highest among all the bidders, the bidder loses the auction with no cost.

Hence, the system reward, R, and cost, C, are given by

$$R(\theta, a) = \theta \sum_{\delta=0}^{a} p_{MP}(\delta|\theta) \tag{3}$$

$$C(\theta, a) = \sum_{\delta=0}^{a} \delta p_{MP}(\delta|\theta) \tag{4}$$

where p_{MP} is the pdf of the market price that can be derived from historical data. Since CTR is a continuous value ranging from 0 to 1, it is discretized into bins and δ denotes the market price of a bin of CTR. The $R(\theta, a)$ represents the probability of winning an auction by bidding a multiplying the probability of getting a click after winning the auction. The $C(\theta, a)$ represents the expected cost of winning an auction by bidding a.

Our objective is to maximize the expected reward while keeping the expected cost below a certain threshold, V. We interpret V as the maximum of the average cost per impression each bidder is willing to spend. The derived policy from CMDP determines the bid price to set in each state.

3.3 Learning from Historical Data: Batch-CMDP

In the CMDP model, the correlation between the CTR and the real feedbacks (clicks of impressions) in the historical data is neglected. We argue that it should

be utilized as valuable experience to learn from. We thus leverage *Batch Reinforcement Learning* (Batch-RL) to derive the best policy from a set of priori-known transition samples [15]. The objective of Batch-RL is to derive a model reflecting the reality learned from the historical data. The advantage of such approach is the efficiency in the learning process compared to the model free approaches, like the Q learning algorithm [19], which needs a huge amount of interactions with the environment to converge to the optimal solution, and which is often not possible in real life applications.

We modify the previous RTB model to derive the best policy from historical data. In the CMDP model, the only variable not derived from historical data is the probability for a click, θ, used in calculating the reward in Eq.(3). We adopt the reward function from the previous model to use only information from the historical data. For each bin of θ, we calculate the corresponding probability of a click using historical data. We denote $f(\theta)$ as the probability of a click given θ. We call this new model *Batch-CMDP*, formally defined as:

$$R(\theta, a) = f(\theta) \sum_{\delta=0}^{a} p_{MP}(\delta|\theta) \tag{5}$$

$$C(\theta, a) = \sum_{\delta=0}^{a} \delta p_{MP}(\delta|\theta) \tag{6}$$

3.4 Market Price Distribution

The market price can be seen as drawing from an unknown distribution generated from the online marketplace. In [7,10], the authors directly model the market price distribution. However, since the winning probability also relies on the CTR estimation, we introduce the correlation between the winning price and the CTR in the estimation of market price distribution. We estimate the probability distribution function of the market price p_{MP} using Eq.(7). This derives implicitly from the joint distribution of the winning price and corresponding CTR, as well as from the distribution of the predicted CTR according to the Bayesian theorem [6].

$$p_{MP}(\delta|\theta) = \frac{p(\delta,\theta)}{p(\theta)} \tag{7}$$

In order to validate our approach, we prove that a strong correlation exists between the market price and the CTR. A commonly used method for this purpose is to calculate the Pearson's correlation coefficient [1]. This technique is efficient in linear correlation cases, however it fails to capture non-linear relationships. *Mutual Information* (MI) [5] is one of the measures that captures any type of non-linear dependencies between two random variables. MI quantifies the amount of information obtained about one random variable given another random variable. In other words, it measures the degree of uncertainty of one

variable knowing the other variable. Formally, the mutual information of two random variables X and Y is defined as

$$MI(X;Y) = \int_Y \int_X p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy \tag{8}$$

where $p(x,y)$ is the joint probability density of X and Y, $p(x)$ and $p(y)$ are the probability density function of X and Y respectively. In the following section, we present the mutual information between the market price and the CTR.

4 Experiment and Results

We have implemented a CMDP model for bidding trained on two real-world RTB datasets. The bidding results are compared with several state-of-the-art bidding algorithms. In this section, we elaborate the experiments and discuss the results.

4.1 Datasets and CTR Prediction

In our experiments, two real-world datasets are used. Due to privacy reasons, the public dataset of RTB bidding logs is very limited. A detailed RTB dataset was released by iPinYou, a leading RTB company in China, for a bidding competition in 2013. This is the only public dataset which contains the historical market price. It contains 19.5M impressions, ~15 K clicks and 1.2 K conversions over 9 ad campaigns. The training and test data are chronologically split into 7 days and 3 days, respectively. The other dataset is from OLAmobile, a global mobile advertising company in Luxembourg. The data are collected from 8 campaigns over 6 days, which include 800 K impressions and 6 K clicks.

The iPinYou dataset contains bid requests, winning impressions, ad clicks, and conversions. Each row in the log file represents a bid request at a certain time. The features can be categorized as user profile, publisher, and ad description. The user profile includes the *time stamp of the visit, user agent, IP address, region*, and *city*. The publisher is represented by *domain and ad exchange ID* and the ad slot is described by *slot size and format, advertiser ID, and creative ID*.

We applied the data pre-processing procedure[2] used in [22], which utilizes the one-hot-encoding method to convert the categorical features into binary features and we used the logistic regression training like in [21] to estimate the predicted CTR.

For the reproducibility, our code is available online[3]. We mainly report and publish the results on the iPinYou dataset. Due to the privacy reason, the OLAmobile dataset is not released, but the results are listed as supplementary.

[2] http://data.computational-advertising.org/.
[3] https://github.com/manxing-du/cmdp-rtb.

4.2 The Correlation Between Market Price and CTR

As introduced in Sect. 3.4, Table 1 shows the results of the normalized mutual information of δ and θ calculated in the iPinYou dataset. The normalization of the Mutual Information scales the results between 0 and 1 where 0 means no relationship and 1 means perfect correlation. It can be inferred from Table 1 that for all campaigns in the iPinYou, $MI(\delta, \theta)$ is significantly higher than 0. We conclude that δ and θ are strongly dependent on each other in the iPinYou data. This supports the rationale of our approach to model the relationship between δ and θ as a batch CMDP.

Table 1. Mutual information of the market price δ and CTR θ

iPinYou Camp	1458	2259	2261	2821	2997	3358	3386	3427	3476	OLA Camp	1	2	3	4	5	6	7	8
$MI(\delta, \theta)$	0.50	0.58	0.59	0.55	0.55	0.56	0.50	0.51	0.53	$MI(\delta, \theta)$	0.18	0.22	0.40	0.35	0.41	0.16	0.36	0.44

4.3 Evaluation Methods

The evaluation of the bidding functions is carried out on a per campaign basis. In our experiments we only focus on the total number of clicks as the KPI, due to the insufficient number of conversions. In addition, since every campaign has a limited budget, our goal is to maximize the number of clicks given the budget constraint. Thus, the *expected cost per click* (eCPC) is also used to measure how efficient the budget is spent.

4.4 Experiment 1: Compare Bid Prices to the Historical Data

In experiment 1, each bidding function computes a bid price for the same bid request and the price is compared with the historical market price. If the bid price is higher than the historical market price, then it wins the auction and the cost is the market price. The subsequent click will be accumulated. Otherwise we assume that the auction is lost with no additional cost. We used the source code available[4] for the work in [7] to generate results for the *Mcpc*, *Lin*, and *RLB* functions.

- **Mcpc**. Sets a maximum eCPC which is the goal of the bidding function. The bid price is calculated by multiplying the max eCPC from the training data with the predicted CTR.
- **Lin**. As proposed by [17], the bid price depends linearly on the predicted CTR as $b_0 \frac{\theta(x)}{\theta_{avg}}$,where b_0 is tuned as in [7] and θ_{avg} is the average CTR in the training set.
- **RLB**. One of the recent papers in [7] formalizes the bidding problem into a reinforcement learning framework. More details in this approach can be found in their paper [7].

[4] https://github.com/han-cai/rlb-dp.

- **CMDP**. Our proposed model of CMDP as described in Sect. 3.2
- **Batch-CMDP**. The second model we proposed in Sect. 3.3 where the policy is learned by using the real feedback (click or not) as the reward.

We first compare each bidding strategy with limited budget. We determine the budget $B = CPM_{train} * c0 * N_{test}$, where $c0 = 1/32, 1/16, 1/8$, and $1/4$. The $c0$ setting is as the same as in [7] to make our results comparable with theirs. In Table 2, the total number of clicks and the eCPC for $c0 = 1/32$ are listed. We find that (i) CMDP and batch-CMDP models outperforms all the other bidding strategies in terms of number of clicks when the CTR estimation has higher AUC, since the state only contains the CTR which directly impacts the performance of our model. (ii) In terms of eCPC, the CMDP solution does not always achieve the least cost. This is due to CMDP trying to keep the cost per impression (CPM) under the averaged value in the training set while obtaining the maximum number of clicks. We did not directly set the goal as the cost per click because before getting clicks, we need to win a certain amount of impressions first. As we can see in Table 2, for example, the *Lin* function has lower eCPC for campaign 2997 while the number of clicks is fewer than *CMDP*. We should note that even all the bidding strategies has the same budget setting, each of them spends different amount of the budget until the end of the test. In other words, within the same budget limit, different algorithms has won different number of auctions. The results suggests CMDP set the bid price efficiently to cover a wider range of impressions and also gets more clicks.

Table 2. Total number of clicks and (eCPC), c0 =1/32

iPinYou Camp	AUC	CTR	Mcpc	Lin	RLB	CMDP	Batch CMDP
1458	97.95%	0.084%	392(3.34)	464 (**1.09**)	424 (3.09)	**464** (2.71)	462 (2.8)
2259	67.12%	0.031%	10 (120.63)	7 (173.52)	12 (101.02)	**13**(**89.47**)	10 (119.16)
2261	62.69%	0.028%	7 (137.06)	9 (105.67)	**11** (**87.39**)	8 (118.10)	7 (116.46)
2821	61.28%	0.057%	17 (107.63)	40 (40.26)	**47** (**39**)	39 (45.32)	41 (45.05)
2997	60.79%	0.34%	62 (4.9)	64 (**2.73**)	**82** (3.7)	71 (2.95)	71 (2.98)
3358	97.48%	0.086%	180 (4.75)	189 (3.77)	199 (4.29)	**208** (**3.38**)	203 (4.28)
3386	77.39%	0.082%	56 (23.12)	55 (**5.52**)	61 (21.21)	**92** (12.99)	91 (14.26)
3427	97.23%	0.068%	227 (5.91)	203 (6.55)	261 (5.14)	**292** (4.47)	292 (**4.36**)
3476	95.88%	0.055%	101 (12.76)	162 (**5.92**)	131 (9.87)	181 (7.16)	**188** (6.82)

Figure 3 illustrates the performance of the bidding functions with respect to different budget settings. The bidding functions are compared in terms of (i) number of clicks (ii) Winning rate (iii) eCPC and (iv) CPM

Without budget control, the *lin* and *mcpc* bidding wins fewer auctions and obtains fewer clicks than all the other bidding functions. Not surprisingly, with the same amount of budget, they win the auctions with high market price, so that the eCPC and CPM are both higher than the others. In general, *RLB*, *CMDP*, and *Batch-CMDP* perform better and especially with the low budget setting, *CMDP* outperforms all the other functions.

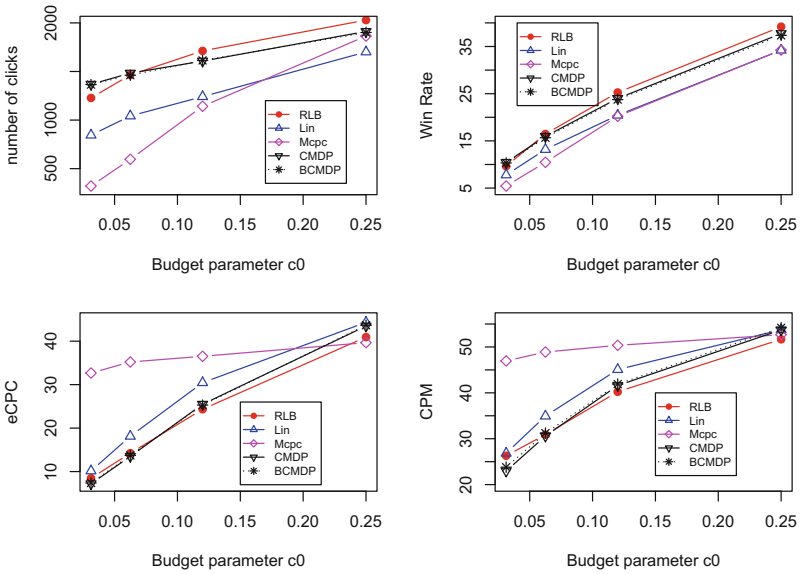

Fig. 3. Overall bidding performance on iPinYou Data.

In Table 3, the results of OLAmobile dataset show that *Batch CMDP* obtains the most clicks among all the bidding functions. In *Batch CMDP*, the reward is computed by Eq.(5) in which the probability of a click is derived from the historical clicking probability given the state θ (pCTR). Since the OLAmobile dataset spans 1–2 days, the conditional distribution of the winning probability given the predicted CTR is more reliable to be used as a factor in the reward function. Thus, it shows that if the model of the environment reflects the reality, *Batch-CMDP* provides the optimal policy for making bidding decisions. In the iPinYou dataset, the training data are from 7 days and the test data are from the

Table 3. Total number of clicks and (eCPC), c0 =1/32

OLA Camp	AUC	CTR	Mcpc	Lin	RLB	CMDP	Batch CMDP
1	69.96%	0.445%	1(46.41)	3(13.63)	3(15.42)	0 (NA)	**4**(30.59)
2	56.79%	0.466%	2(45.67)	2(39.56)	2(45.23)	3(29.84)	**4**(26.51)
3	73.22%	1.827%	**6**(2.74)	5(1.21)	6(2.51)	2(2.26)	3(1.53)
4	75.18%	0.938%	21(6.04)	22(4.34)	**26**(4.86)	14(8.94)	26(6.82)
5	71.35%	1.833%	3(3.4)	3(3.39)	4(2.54)	4(2.23)	**13**(2.64)
6	58.15%	0.465%	8(30.01)	16(14.28)	10(23.9)	6(39.55)	**28**(38.54)
7	67.69%	1.237%	4(60.95)	14(16.53)	5(48.42)	9(27.21)	**23**(23.83)
8	68.07%	0.554%	33(9.46)	51(5.78)	61(5.07)	71(4.37)	**88**(5.93)

following 3 days. Our interpretation is that the market price model changes over time, thus the model based on the long term history degrades the performance of *CMDP* and *Batch-CMDP*.

We should also note that in Eq. 1, the sum of $\rho(s, a)$ over the entire state and action is 1. In other words, the policy learned by CMDP also depends on the pCTR distribution in the training set. If the pCTR distribution in the test set changes dramatically, the policy may lead to budget overspending or underspending. The dynamics of the pCTR distribution is a strong focus of our future work.

4.5 Experiment 2: Compare Bidding Functions in the Same Environment

In the previous experiment, the performance of the bidding functions is independently compared with the historical winning prices. However, in a real world scenario, every bidder will try to improve his/her bidding functions at any time. Thus, we present the impact of the fluctuation of the market price distribution on the bidding strategies. We simulate the scenario by assuming that the historical winning prices come from a single virtual bidder and let the other bidding functions compete with each other as well as with the virtual bidder. The winner is the one which bids the highest and the winning price is the second highest price. If more than one bidders set the same price, all of them win the auction, which produces the maximum number of auctions and clicks each function can win in this setting.

If more than one functions bid the same price, all of them are considered as winners. The corresponding clicks and impressions are denoted as *Dual_win_clk* and *Dual_win_imp* respectively in Table 4. Meanwhile, the single winner case is represented as *Sig_win_clk* and *Sig_win_imp*. The *RLB*, *CMDP*, and *Batch*

Table 4. Comparing bidding functions in the same environment, c0 = 1/32

Camp. 1458	mcpc	Lin	RLB	CMDP	Batch CMDP
Dual_win_clk	55	367	361	101	100
Sig_win_clk	9	46	28	2	14
Dual_win_imp	216	700	2978	3512	3350
Sig_win_imp	30991	84	13731	4755	41783
total_ecpc	23269	453	2608	4647	10059
Camp. 2997	mcpc	Lin	RLB	CMDP	Batch CMDP
Dual_win_clk	0	0	7	5	3
Sig_win_clk	45	0	48	4	10
Dual_win_imp	11	0	2753	2736	2369
Sig_win_imp	15560	0	25799	1051	3718
total_ecpc	7458	0	5441	3624	4077

CMDP models are trained using the historical market price and the experiment was running on the test data. In this setting, the market price in the test set shifts towards higher prices when more than one functions bid higher than the historical market price.

The result suggests that for the campaigns with high AUC (e.g. campaign 1458), the linear bidding function targets the right impressions to bid high. Other functions, for example, like *CMDP* wins 10 times more impressions but only get 1/3 of clicks as *Lin*, which significantly increase the eCPC. On the contrary, *Lin* loses its advantage when the predicted CTR is not accurate since it only relies on pCTR to calculate the bid price. One extreme case is campaign 2997 having the lowest AUC in the dataset. *Lin* sets the bid price too low comparing to other functions and it does not win any impression. The results also show that the eCPC should not be the only metric to evaluate how well the bidding function performs. For example, for campaign 2997, *CMDP* has a lower eCPC while having 6 times less number of clicks than *RLB*. In this case, *CMDP* bids more conservative than *RLB* since *CMDP* follows the policy learned from the pCTR density function in the training set.

5 Conclusions and Future Work

In this paper, we formalize the bidding problem in the RTB system as a Constrained Markov Decision Process. We use linear programming to maximize the total reward with a cost limit. The reward is either derived from the CTR estimation (in CMDP) or from the historical observations (in Batch-CMDP), in which case the best policy is learned given the training data. We use Bayesian inference to obtain the market price distribution, which not only considers the correlation between the market price and the state (pCTR) but also captures the dynamics of market price. Our model outperforms the state-of-art bidding functions in terms of the total number of clicks constrained to a limited budget. However, when the bidding functions compete with each other, linear bidding performs the best for campaigns with a high AUC while RLB obtains more clicks for campaigns with a low AUC. *CMDP* relies on the correlation of the historical market price distribution and the predicted CTR distribution, thus bids more conservative compared to the others.

For future work, we will model the time-dependent dynamics of RTB to improve on the use of a single fixed market price distribution. In addition, we will investigate a model-free approach which does not assume a modeling of the market price distribution but only learns from the rewards.

Acknowledgement. We sincerely thank Prof. Weinan Zhang and his research group from Shanghai Jiaotong University for the short visit. Manxing thanks the National Research Fund (FNR) of Luxembourg for the research support under the AFR PPP scheme and thanks Dr.Tigran Avanesov from OLAmobile for the feedback.

References

1. Aggarwal, C.C.: Data Mining: The Textbook. Springer, Cham (2015)
2. Altman, E.: Constrained Markov Decision Processes. CRC Press, Boca Raton (1999)
3. Amin, K., Kearns, M., Key, P., Schwaighofer, A.: Budget optimization for sponsored search: censored learning in MDPs. In: Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence. AUAI Press (2012)
4. Amin, K., Kearns, M., Key, P., Schwaighofer, A.: Budget optimization for sponsored search: censored learning in MDPs. CoRR (2012)
5. Applebaum, D.: Probability and Information: An Integrated Approach, 2nd edn. Cambridge University Press, Cambridge (2008)
6. Barber, D.: Bayesian Reasoning and Machine Learning. Cambridge University Press, New York (2012)
7. Cai, H., Ren, K., Zhag, W., Malialis, K., Wang, J.: Real-time bidding by reinforcement learning in display advertising. In: Proceedings of the 10th ACM International Conference on Web Search and Data Mining (WSDM) (2017)
8. Chakrabarti, D., Agarwal, D., Josifovski, V.: Contextual advertising by combining relevance with click feedback. In: Proceedings of the 17th International Conference on World Wide Web (WWW) (2008)
9. Chen, Y., Berkhin, P., Anderson, B., Devanur, N.R.: Real-time bidding algorithms for performance-based display ad allocation. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2011)
10. Cui, Y., Zhang, R., Li, W., Mao, J.: Bid landscape forecasting in online ad exchange marketplace. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2011)
11. Geibel, P.: Reinforcement learning for MDPs with constraints. In: European Conference on Machine Learning (2006)
12. Ghosh, A., Rubinstein, B.I., Vassilvitskii, S., Zinkevich, M.: Adaptive bidding for display advertising. In: Proceedings of the 18th International Conference on World Wide Web, pp. 251–260. ACM (2009)
13. Hoelzel, M., Ballvé, M.: The programmatic-advertising report: mobile, video, and real-time bidding drive growth in programmatic. BI Intelligence (2015)
14. Krishna, V.: Auction Theory. Academic Press, San Diego (2009)
15. Lange, S., Gabel, T., Riedmiller, M.: Batch Reinforcement Learning. Springer, Heidelberg (2012)
16. Liu, C.: US Ad Spending: eMarketer's Updated Estimates and Forecast for 2015–2020. Industry report (2016)
17. Perlich, C., Dalessandro, B., Hook, R., Stitelman, O., Raeder, T., Provost, F.: Bid optimizing and inventory scoring in targeted online advertising. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2012)
18. Schwartz, E.M., Bradlow, E., Fader, P.: Customer acquisition via display advertising using multi-armed bandit experiments. Ross School of Business Paper (2015)
19. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, vol. 1. MIT Press Cambridge, London (1998)
20. Xu, J., Lee, K.c., Li, W., Qi, H., Lu, Q.: Smart pacing for effective online ad campaign optimization. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2015)

21. Zhang, W., Yuan, S., Wang, J.: Optimal real-time bidding for display advertising. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2014)
22. Zhang, W., Yuan, S., Wang, J.: Real-time bidding benchmarking with iPinYou dataset. CoRR (2014)