COMPGW02 Web Economics Coursework Project

Individual Report

Tim Warr - 13026783
ucabtw2@ucl.ac.uk

ABSTRACT

In this report, we explore the problem of real-time bidding in display advertising, using the iPinYou dataset. We develop a bidding algorithm that combines CTR estimation (calculated using gradient-boosted decision trees) and bid-landscape forecasting (via a mixture of Gaussians) to achieve results superior to the linear bidding strategy. These components are later used as part of the group report, in which we develop a multi-agent reinforcement learning bidding strategy.

Keywords

Real-time bidding, demand-side platform, CTR prediction, bid landscape forecasting, reinforcement learning

1. INTRODUCTION

This report forms the individual contribution to COMPGW02 Web Economics coursework project. Submitted separately is a group report. The corresponding code for this project (both individual and group) can be found on our GitHub. [1]

1.1 Problem Context

The majority of online display adverts are now served through real-time bidding [1] in which an advert slot is auctioned in real-time for each user visit. This has a number of advantages over the more commonly-known *keyword advertising* in which an advertiser pays to have their ad appear in the results when a user types a particular phrase to search the Web. The most obvious of which, is that it allows advertisers to obtain better control over which users they pay to show their adverts to.

Each time a user visits a website, they reveal information about themselves, such as their operating system and IP address. User information is combined with statistics about the ad slot (such as location, size and format) to form a feature vector. To place a bid correctly, advertisers must devise machine learning algorithms that can determine whether a given impression is likely to result in a click.

For a given impression, each advertiser submits a bid for the price they are willing to pay (in real-time; the whole process takes less than 0.1 seconds). The advert belonging to the highest bidder is shown to the user. The auction is a second-price auction meaning that the highest bidder pays the second-highest price.

[1]`https://github.com/uclwe/rtb`

In this report we develop some bidding strategies to perform well in the real-time bidding for display advertising task.

1.2 Approach

The report is structured as follows: in Section 2, we briefly explore some of the related work in the domains of CTR-prediction, bid-landscape-forecasting and bid optimisation. In Section 3 we perform an in-depth analysis of the dataset provided and describe our individual bidding strategies. This is followed by a summary of our results, comparing the performance of multiple strategies, and finally brief summary of what we would like to work on with more time.

2. RELATED WORK

In [7], the authors provide a basic statistical analysis of the full iPinYou dataset (note that in this project we only use a smaller subset - due to computational complexity and resource limitations). They also provide some benchmarking algorithms for CTR-prediction (logistic regression & gradient boosted decision trees) and bid optimisation (constant, random, linear, bidding below max eCPC).

The authors of [3], introduce a decision tree + logistic regression model to predict clicks on Facebook adverts. They highlight the importance of good feature selection and describe one way to sensibly down-sample and recalibrate a high dimensional dataset into one that is less computationally intensive to perform inference over.

Other papers focus on the problem of bid-landscape forecasting, which involves predicting the market price distribution of a give impression. Specifically [2] forecasts the bid for each sample using GBDT regression and then fit a GMM to generate a campaign level bid distribution.

Non-linear bid optimisation strategies are described in papers such as [5], [6]. More specifically, in [6], the authors describe a strategy which takes into account the budget and the campaign lifetime, wanting to bid on more impressions instead of a subset of high valued ones. This strategy is cost effective as it bids on lower valued impressions, meaning that they save money and the chances of winning are higher. [5], the authors challenge the idea of using Value based bidding, as they suggest that this strategy favours users with an already high action rate (e.g. click, or conversion). Instead they suggest a lift-based bidding strategy that is based on the action rate lift i.e. focusing on users that would be more influenced by the ad specifically.

The authors of [1], [4] formulate the real-time bidding problem as a Markov Decision Process (MDP) and introduced model-based and model-free (respectively) reinforcement learning algorithms to develop bidding strategies that are able to additionally consider budget constraints.

3. APPROACH & RESULTS

In this Section, we provide answers to the question asked in Section 3 of the assignment.

3.1 Data Exploration

First we explore the subset of the iPinYou dataset, provided for this assessment.

3.1.1 Basic Statistical Information

Please see Table 1 for a breakdown of the basic statistical information regarding the Train, Validation and Test sets (such as number of impressions, CTR, aCPM etc - we define what each of these metrics mean in the group report). We see that the validation and test sets are of equal size and contain approximately one eighth of the number of impressions as in the training set. The metrics defined are fairly consistent across the train and validation sets (see Table 1), but we find that the average cost per click in the validation set is approximately 10% higher than that of the training set. Our budget is 6250 CNY, which is approximately equal to 1/4 of the total cost of the validation set.

3.1.2 Feature Overview

From here onwards, the statistics will refer only to the training set only (unless stated otherwise). We find that there are 22 features included in the dataset, including user information (such as their region and operating system); advert information (including advertiser and ad exchange) and general information (such as time and day details).

We calculate the "uniqueness" of each feature (i.e. the number of possible values it can take divided by the number of impressions) and find that *bidid* is unique, *userid* is near-unique (96%) and *url* and *IP* are over 20% unique. A unique or near-unique feature is unlikely to offer much information when trying to perform inference. Similarly we note that *urlid* has only 1 possibility - so it doesn't offer information that could be used for inference.

We now explore the distribution of clicks and CTR across a few of the features (see GitHub for a more comprehensive breakdown).

- Device type (Figure 1)

We note that *"useragent"* can be separated into *device* and *OS*. We see that the CTR for Android and iOS is considerably higher than the other operating systems. In [7] they put this down to the "fat finger effect".

- Ad exchange (Figure 2)

We see that the CTR for the "null" ad-exchange is far higher than that of the labelled ones. In fact it contains over 12% of the clicks and only 2% of the impressions.

- Hour (Figure 3)

We plot the volume of bids attributed to each hour of the day and observe a clear distinction between day time and night time. We observed a higher CTR between the hours of 5pm and midnight.

- Usertag (Figure 4)

There are 69 unique usertags that appear in the train set (all appear in validation and test too). Of these we note that usertag 10063 corresponds to less than 0.2% of the impressions, but over 10% of the clicks.

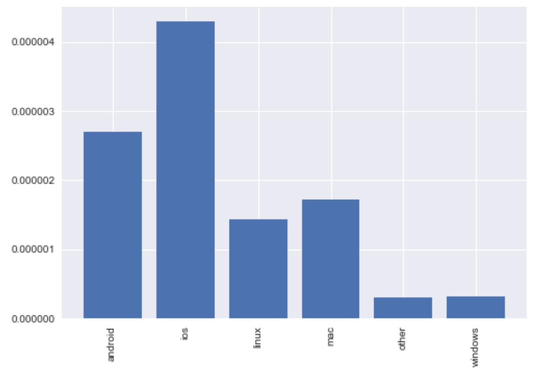

Figure 1: CTR across Device Types

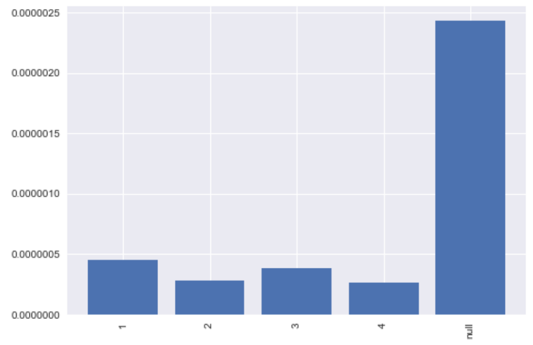

Figure 2: Adexchange CTR

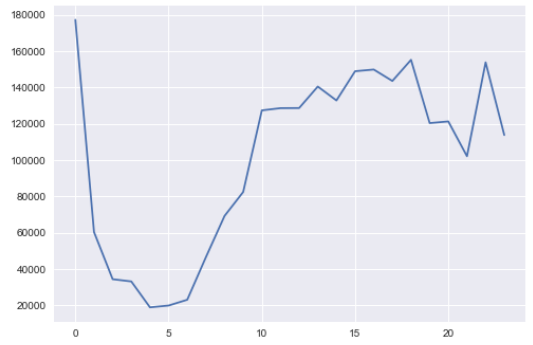

Figure 3: Hour vs Volume

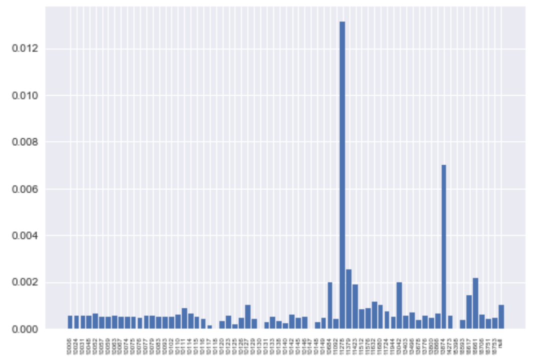

Figure 4: Usertag CTR

Table 1: Basic Statistical Information

	Impressions	Clicks	Cost	CTR	avgCPM	eCPC
Train	2.43×10^6	1793	1.90×10^5	7.38×10^{-4}	78.151	105.96
Valid	3.04×10^5	202	2.38×10^4	6.65×10^{-4}	78.234	117.91
Test	3.03×10^5	-	-	-	-	-

3.1.3 Further Analysis

In Figure 5, we plot the log distributions of the market price in the train and validation set. We see that their distributions are visually very similar and that they could possibly be well represented by a mixture of Gaussians model (see Section 3.4.2).

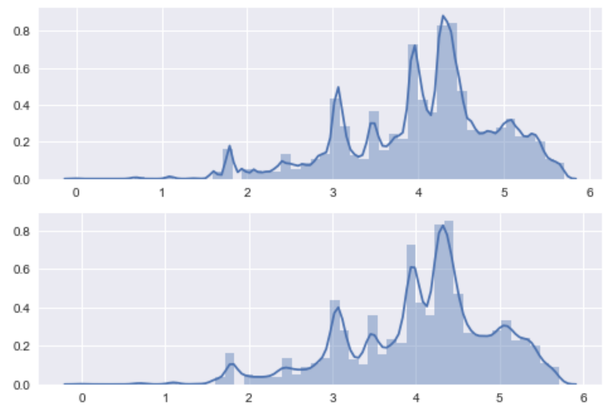

Figure 5: Log(1+payprice)

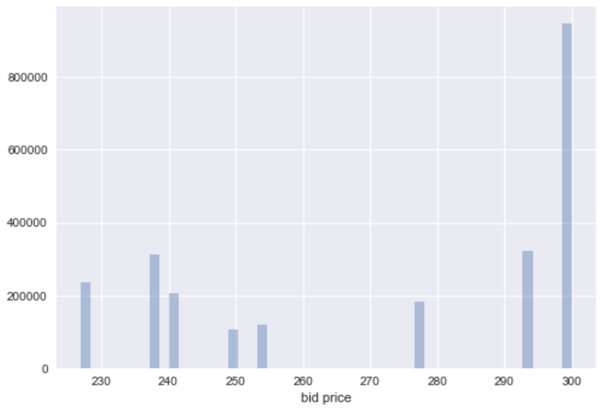

Figure 6: Bid price distribution

Figure 6 shows the distribution of bid price for the training set. We observe that the bidding strategy appears to be one which contains only 8 bid values (ranging from 227 to 300).

We find that on average (mean 195, median 209) the bid prices are considerably higher than the market prices.

3.2 Basic Bidding Strategies

PLEASE REFER TO GROUP REPORT.

3.3 Linear Bidding Strategy (LIN)

PLEASE REFER TO GROUP REPORT.

3.4 Best Bidding Strategy

As shown in Figure 1 (below), a bidding strategy can be split in to three main components (shown in red): CTR Estimation, Bid Landscape Forecasting and Bid Optimisation. In this subsection, I outline my approach to these three components.

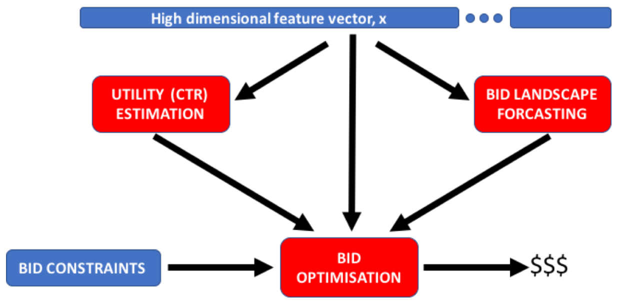

Figure 7: General bidding strategies

3.4.1 CTR Estimation

CTR estimation is the task of predicting the probability that an impression will result in a click. Due to the innate class-imbalance of the domain, this task is non-trivial.

For this, we trained three separate classifiers to predict click-through-rate (logistic regression (LR), gradient boosting decision trees (GBDT) and a multi-layer-perception (MLP)). The parameters for the LR and GBDT were chosen using random grid search, whilst due to time constraints, the MLP architecture and parameters were left untuned, based primarily on a Kaggle kernel that aimed to spot fraudulent transactions [2].

The performance of each classifier was assessed based on cross-entropy loss and ROC-AUC on the validation set (shown in Table 2). In Figures 8 and 9, we see the distribution of CTR prediction for the LR and GBDT models on the validation set. Interestingly, we see a drastic difference between the type of predictions each model makes. GBDT assigns a very high probability to low CTR, whereas the LR predictions are much less conservative.

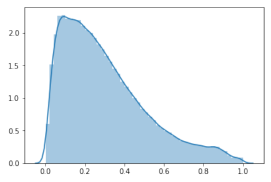

Figure 8: LR

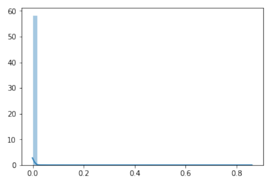

Figure 9: GBDT

[2]https://www.kaggle.com/randyrose2017/using-scikit-learn-and-keras-for-fraud-detection

Table 2: Cross-entropy loss of CTR predictors

Classifier	Log Loss	ROCAUC Score
LR	0.43892	0.85063
GBDT	0.00409	0.87417
MLP	0.41010	0.82766

3.4.2 *Bid Landscape Forecasting*

This is the problem of market-price estimation. The first approach taken was a naive one and involved training regression models to predict the market price of each individual impression, with performance evaluated using mean-squared error. We found that these regressions resulted in poor predictive-performance, probably due to the variance in market price even for an individual impression.

A better approach is to instead consider that the market price for each impression is drawn from a probability distribution. This is challenging because we are effectively trying to predict the distribution for each feature vector based on only one sample. If we knew the possible distributions that each impression could belong to, then it would be possible to train a classifier to predict the distribution that the impression belongs to.

If the market price PDF, given x is $m(\delta, x)$, then its CDF is the winning probability given each specific bid price. Based off the observation that the log market price could feasibly be drawn from a mixture of Gaussian distributions (see Figure 5), we followed an approach similar to [2] and fit a mixture of Gaussians to the pay price data. We denote the number of Gaussian distributions in the GMM as N.

Given the N Gaussian distributions, we can then consider the classification task of matching a given impression to the distribution log normal distribution which was most likely to generate its marker price. Following the success of the GBDT model in the CTR estimation phase, and the fact that the authors of [2] also used GBDT for their distribution classification, this is the classifier that we implemented.

When choosing N, there is a clear trade-off between the extent to which a GMM can fit the log market price and the performance of the distribution classifier. To highlight two extremes, we note that if $N = 1$, the single Gaussian distribution does not fit the log market price well, but classification is trivial. Conversely, if N is large, then the GMM could near-perfectly describe the market price distribution, but the classification task will be infeasible. We found that $N = 4$ provided a solid medium between the need to fit the price data well and easy of classification. Our 4-GMM distribution is shown in Figure 10, below - this should be compared to the true train and validation distributions in Figure 5.

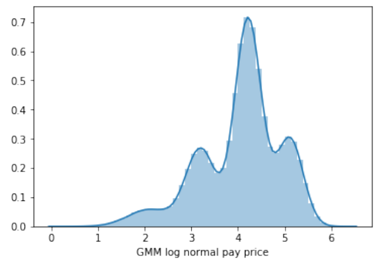

Figure 10: GMM log price distribution

3.4.3 *Bid Optimisation*

Bid Optimisation involves combining the CTR estimation with the market-price estimation, as well as the possible addition of constraints (such as the remaining budget and number of impressions).

Below we outline four of the approaches taken to develop bid optimisation strategies.

NAIVE NON-LINEAR (NNL)

We initially tried naive non-linear bidding strategies (using the CTR predictors mentioned previously) such as:

$$bid(x) = \beta_1 pCTR + \beta_2 pCTR(x)^2 \qquad (1)$$

$$bid(x) = \beta_3 \mathrm{e}^{pCTR(x)} \qquad (2)$$

$$bid(x) = \beta_4 \log(pCTR(x) + 1) \qquad (3)$$

We optimised the parameters β_i using the validation set, with the condition that $\beta_2 > 0$ (else this would be equivalent to linear bidding). The strategy in equation (1) performed best for small β_2, tending towards the linear strategy. Equation (2) method emphases the effect of the CTR predictions and was detrimental to the performance. This is because the CTR predictions are not perfect, so any errors would be exaggerated. As a result of this, we tried the equation in (3), which dampens the effect of the CTR predictions. Out of the three naive methods tried, this was the strongest (but still inferior to liner method) - a comparison is shown in Table 3.

LINEAR VARIANT (LV)

We also implemented a slight variant of the linear bidding strategy, which consisted of four linear bidding strategies corresponding to the four distributions in the GMM. The distribution of a given impression was predicted, the base-bid for that distribution was used.

$$bid = \alpha_i \times pCTR \qquad (4)$$

Whilst this strategy is promising, and is guaranteed to be *at least as good as LIN* (since setting $\alpha_i = base\_bid(LIN)$ is equivalent to LIN), it comes with a drawback. The search space for base bids increases exponentially in the number of distributions. This made it infeasible to grid search across the best base bids in a reasonable amount of time. For this reason, we performed a random search of 1,000,000 quadruples $(\alpha_0, \alpha_1, \alpha_2, \alpha_3)$ only. The performance was strong (159) clicks, but still inferior to LIN. With a more extensive search (or efficient) search of the parameter space, we would expect this strategy to overtake LIN. *Note - we did not see this approach mentioned in literature.*

INVERSE CDF (ICDF)

We previously described that our most successful method for finding the distribution of a given impression was to fit a Gaussian mixture model of the log price and then use GBDT classification. We also recall that the CDF of a bid's market price distribution gives the winning probability at each price. In Figure 11, we show the probability of winning an auction for each distribution against the market price that one would have to pay to win with that probability.

If one could derive a function that chooses the probability of wanting to win a given impression, then they could simply

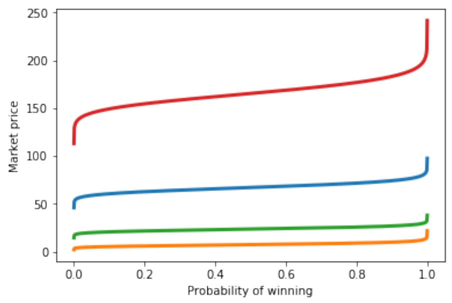

Figure 11: Inverse CDF

"look up" the amount to bid according to the inverse CDF. We developed and tested four of these functions:

$$p_{win}(x) = \omega_0 \tag{5}$$

$$p_{win}(x) = \omega_1 \times pCTR(x) \tag{6}$$

$$p_{win}(x) = \omega_2 \times log(1 + pCTR(x)) \tag{7}$$

$$\begin{aligned} p_{win}(x) &= 0.9999, && \text{if } pCTR(x) \geq \omega_3 \\ p_{win}(x) &= 0.5, && \text{if } pCTR(x) \geq \omega_4 \\ bid(x) &= 0, && \text{otherwise} \end{aligned} \tag{8}$$

.

If $p_{win}(x)$ exceeded 1 in any of the calculations above, then it was replaced with 0.9999 (else the bid would have been ∞). The first strategy alone was essentially a variant of constant bidding (where a different constant amount was bid for each distribution). In equation (5), we wanted to explore whether a linear relationship between CTR and desire to win was appropriate. In strategy (6) we wanted to decrease the effect of CTR on the desire to win, when compared to (7). All strategies performed equal to or better than the linear bidding strategy, with the best (6) obtaining 168 clicks on the validation set. *Note - we did not see this approach mentioned in literature.*

LONG-TERM BIDDING (LTB)

We explored the effect of applying an "aggression" transform to the bids from the linear bidding strategy. The idea behind this was to see the effect of reducing the price of early bids, so that higher bids could be placed for impressions later on. *Note - we did not see this approach mentioned in literature.*

3.4.4 Results

Throughout the project, we were impressed at just how strong the linear bidding benchmark could perform. This meant that it was no mean feat trying to outperform its results.

We see in Table 3 that ICDF strategy achieved the highest number of clicks and best CTR out of the methods we attempted. This highlights the fact that successful bid landscape forecasting can have a positive influence on a bidding strategy.

We also saw the importance of considering constraints (in LTB the number of impressions remaining) when designing a successful strategy. With only a slight variation of the linear bidding strategy, we were were able to increase the number of clicks to 164, but do so using only 87% of the budget (when compared to the 98% used by LIN).

Table 3: Comparison of bidding strategies

Approach	Cost	Clicks	CTR (%)	CPC
LIN	98%	163	0.106	37.76
NNL	98%	157	0.114	39.03
LV	98%	159	0.105	38.59
ICDF	99%	168	0.121	36.83
LTB	87%	164	0.115	33.33

3.5 Group Bidding Strategy

PLEASE REFER TO GROUP REPORT.

4. CONCLUSION

In conclusion, we have provided a solid analysis of the iPinYou dataset provided and developed a variety of strong approaches to the RTB problem. We explored variety of linear and non-linear bidding strategies and developed a good-performing market price distribution predictor based on the literature.

We note that the objective of this coursework was to explore bidding strategies that achieved the highest number of clicks only. In reality, there are many more objectives that an advertiser would want to consider such as the number of auctions won (regardless of click-through). Whilst we were unable to *drastically* improve the results achieved by LIN using the market price predictor that we developed - it could have further use when considering other objectives

4.1 Future work

With more time we would like to further explore the effect of bidding under budget constraints.

The group report of this coursework contains information on our attempt at incorporating multi-agent reinforcement learning into the problem. As shown in the literature, reinforcement learning is very helpful when dealing with budget constraints whereas MARL is particularly helpful when trying to better model competition. To this extent, in further work we would like to focus on building more robust RL agents and simulating MARL across many agents.

4.2 Contribution

All team members contributed equally towards the group aspects of this project.

ACKNOWLEDGMENTS

We would like to thank Ying Wen for his provision of the submission API, which was useful in testing, during the development phase of our algorithms.

5. REFERENCES

[1] H. Cai, K. Ren, W. Zhang, K. Malialis, J. Wang, Y. Yu, and D. Guo. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 661–670. ACM, 2017.

[2] Y. Cui, R. Zhang, W. Li, and J. Mao. Bid landscape forecasting in online ad exchange marketplace. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 265–273. ACM, 2011.

[3] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, et al. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, pages 1–9. ACM, 2014.

[4] D. Wu, X. Chen, X. Yang, H. Wang, Q. Tan, X. Zhang, and K. Gai. Budget constrained bidding by model-free reinforcement learning in display advertising. *arXiv preprint arXiv:1802.08365*, 2018.

[5] J. Xu, X. Shao, J. Ma, K.-c. Lee, H. Qi, and Q. Lu. Lift-based bidding in ad selection. In *AAAI*, pages 651–657, 2016.

[6] W. Zhang, S. Yuan, and J. Wang. Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1077–1086. ACM, 2014.

[7] W. Zhang, S. Yuan, J. Wang, and X. Shen. Real-time bidding benchmarking with ipinyou dataset. *arXiv preprint arXiv:1407.7073*, 2014.

6. APPENDIX

6.1 Feature Selection

- The following features were dropped (since they were unique, or near-unique across the training set) *"bidid", "userid", "IP", "url"*
- *"urlid"* was dropped, since it only took 1 possible value in the training set
- *"usertag"* was converted from strings of usertags, to 1-hot-encodings for each of the 69 possible user tags
- The following features were one-hot-encoded *"useragent", "region", "city", "adexchange", "domain", "slotid", "slotvisibility", "slotformat", "creative", "keypage", "usertag", "advertiser"*
- The following features were left unchanged: *weekday, hour, slotwidth, slotheight, slotprice*