ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Time Series Modeling of Market Price in Real-Time Bidding

Manxing Du[1], Christian Hammerschmidt[1], Georgios Varisteas[1], Radu State[1], Mats Brorsson[2], and Zhu Zhang[3] *

1- University of Luxembourg - Luxembourg

2- The Royal Institute of Technology (KTH) - Sweden

3- Iowa State University - United States

Abstract. Real-Time-Bidding (RTB) is one of the most popular online advertisement selling mechanisms. Modeling the highly dynamic bidding environment is crucial for making good bids. Market prices of auctions fluctuate heavily within short time spans. State-of-the-art methods neglect the temporal dependencies of bidders' behaviors. In this paper, the bid requests are aggregated by time and the mean market price per aggregated segment is modeled as a time series. We show that the Long Short Term Memory (LSTM) neural network outperforms the state-of-the-art univariate time series models by capturing the nonlinear temporal dependencies in the market price. We further improve the predicting performance by adding a summary of exogenous features from bid requests.

1 Introduction

In recent years the programmatic trading paradigm has become dominant in the online advertising industry. Unlike in the contextual advertising [1], where the ad is associated with the content of the webpage, or in the sponsored search [2] where the bidding target is the searching keyword, in Real-Time Bidding (RTB) system, the ad impressions are sold individually upon each user's visit. Once a user opens a webpage, the browser sends the user profile combined with the description of the publisher to an Ad Exchange (ADX) asking for a suitable ad. The ADX distributes the requests to different advertisers. Usually, a Demand-side-Platform (DSP) on behalf of advertisers receives such requests and sends a bid price back to the ADX for each ad display opportunity. In RTB, the most popular type of auction is the Vickrey auction, a.k.a the second-price auction [3]. In these auctions, the winner is the one with the highest bid and pays the second highest bid, called the winning price or the market price. In case of losing, the market price is not observable to the bidder. The ADX notifies the winner the market price it needs to pay and the publisher shows the winner's ad on its page. Depending on the advertiser's target, the DSP only gets paid when users click the ad or accomplish certain conversions. By receiving the user's activity from the advertisers, the DSP adjusts its bidding strategy towards targeting the users who generate more profits.

*This research has been supported by the National Research Fund (FNR) of Luxembourg under the AFR PPP scheme

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

The goal of a DSP is to bid strategically and maximize its profits given a limited budget in the highly dynamic bidding environment. A key component in the bidding strategy design is the market price modelling which provides reference for estimating the winning probability of the proposed bid price. The underlying dynamics of the bidding market can be mainly attributed to the hybrid bidding strategies used by all the bidders. In addition, publishers also adjust their pricing model to set a minimum price to pay which is known as a floor price [4]. Therefore, besides other bidders, publishers also optimize their strategies introducing more uncertainties into the bidding environment.

Many studies addressed the bidding strategies optimization problem in RTB [5, 6, 7], in which, the market model is taken from a statistic counting based estimation. The temporal dependency of the market price has been neglected. The authors in [4] show their data has a strong autocorrelation of the market price on an hourly scale. Inspired by this work, and based on our observation of the price dispersion on a per bid request basis, we aggregate bid requests by time and formulate a time series of the averaged market price per aggregated segment. In this paper, we first adopt an univariate LSTM-based Recurrent Neural Network (RNN) to predict the market price, and demonstrate that the volatility of the market price can be implicitly captured by exogenous variables like user features in bid requests.

2 Related Work

Market price estimation is one of the key components of bidding strategies optimization in online advertising. It is straightforward to model the winning probability of a certain bid price by taking the c.d.f. of the market price distribution. Many work in the RTB fields directly define a function to approximate the convex form of the winning probability [5] or adopt the non-parametric Kaplan-Merier estimator to get the market price distribution [8, 9]. For one ad campaign, it may target various user groups and compete with different bidders in every auction. It results in different winning probabilities with the same bid price. However, the temporal dynamics of the market price have been largely neglected in the previous studies. One of these few studies demonstrates how Generalized Linear Models (GLM) can infer the mean shift of the market price as a function of the floor price set by the publisher [4]. Our motivation to address temporal aspects of RTB in this work is motivated by observations in this study.

3 Problem Formulation

In the RTB system, each ad campaign keeps participating in the online auctions until its budget exhausts. In theory, to reach the Nash equilibrium in the second price auction, all the bidders are encouraged to submit their estimation of impression's true value as the bid price [3]. However, in practice, the behaviour of each bidder is highly constraint by its budget setting. Due to the high volatility

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

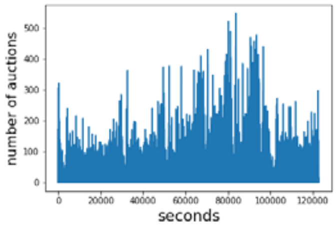

Fig. 1: Number of impressions per second

in the bidding market, predicting the winning price for each individual request can be very challenging. Instead, we propose to model the market price at an aggregate level in the temporal domain.

Figure 1 shows the number of impressions per second from an ad campaign in our dataset. The traffic volume varies across different ad campaigns. Since the number of auctions on different time scale varies over time, to keep the number of auctions in each segment constant, we aggregate auctions by the window size 100, 500, and 1000. The average market price ($\overline{m_t}$) of each window is modeled as a time series.

3.1 Baseline: Autoregressive Model

We take the series of $\overline{m_t}$ and fit the Autoregressive Integrated Moving Average (ARIMA) model as the baseline to forecast the mean market price per window as an univariate model. As the name suggested, an ARIMA model consists of three parts and is specified by the autoregressive order p, the non-seasonal difference order d, and the order of prediction error q. The model is defined as: $y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} - \theta_1 \epsilon_{t-1} - ... - \theta_q \epsilon_{t-q}$, where ϵ is the prediction error at each time step. In the experiment, the parameters are set to be $p, d, q = 1, 0, 0$.

3.2 Recurrent Neural Networks

Recurrent neural networks (RNN) [10] is designed for learning the non-linear long term dependencies in sequential data. As a variant of RNN, the Long Short Term Memory (LSTM) model [11] improves selective recall of useful information in long sequences. The inputs $y_1, ..., y_{t-1}$ are the historical market prices prior to time t.

3.3 Integration of Exogenous Features

The previous two models only focus on modeling the price without considering other variables. However, the information in the bid request is one of the key sources for each bidder to set its bid price. Therefore, we further investigate how the additional information in the bid requests contribute to the price prediction.

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Based on the aggregation setting in this work, we summarize the features from each request into a vector per window to describe the heterogeneous environment. For each feature, we calculate a window-based histogram of the feature values. Non-observed values in a window remain as zero. The histograms of each feature are concatenated into one vector. In this case, we ensure that the dimension of the input vector is the total number of unique feature values and remains the same. In addition, we concatenate a series of market price prior to the prediction time to the feature vector. The historical market price indicates the past trend of the price. The combined input is fed into a LSTM layer and the target value is the averaged market price. We denote the model as LSTM(X).

$$x_t = [\underbrace{0.1, ..., 0.2, ...0}_{city, \sum=1}, \underbrace{0.2, .., 0.4}_{browser, \sum=1}, ..., y_1, ...y_{t-1}]$$

4 Experiments

In this section, we describe the dataset and discuss the results of three experiments based on ARIMA, LSTM, and LSTM(X) models as outlined before.

4.1 Dataset

In our experiments, we use a public RTB dataset. It was released by iPinYou, a leading RTB company in China and contains 19.5M impressions, 15K clicks and 1.2K conversions over 9 ad campaigns. The iPinYou dataset contains winning bid requests with their market price and the feedback from the end users, namely the clicks and conversions.

Each bid request contains features including profiles of the user, the publisher, and the ad slot. The user profile includes *weekday, hour, user agent, IP address, region, city*. The publisher is represented by *domain, url, ad exchange ID* and the ad slot is described by *slot size, format, advertiser, creative ID*. We included all the features above excluding the ones with high variety: IP address and URL. The train and test data are split by the time sequence, as is shown in [12].

Table 1: MSE of ARIMA model

campaign	window size		
	100	500	1000
	MSE	MSE	MSE
3358	159.75	211.07	279.11
3386	158.14	214.14	294.33
3427	176.05	235.17	263.70
3476	190.06	231.96	227.09
1458	153.49	158.94	129.44
2259	185.69	38.86	32.36
2261	174.88	43.10	27.66
2821	346.95	38.92	27.79
2997	76.86	20.35	13.66

Table 2: Comparing ARIMA and LSTM

campaign	window size		
	100	500	1000
	Improves over ARIMA (%)		
3358	-10.28	2.17	5.16
3386	7.02	5.42	13.12
3427	-1.64	4.56	14.02
3476	11.92	36.48	44.12
1458	13.26	24.10	20.56
2259	59.18	7.89	12.06
2261	64.64	37.09	21.73
2821	22.34	-29.43	-42.02
2997	63.26	-3.65	-28.26

Table 3: Comparing LSTM and LSTM(X)

campaign	window size		
	100	500	1000
	Improves over LSTM (%)		
3358	-0.45	27.20	17.79
3386	10.01	63.17	144.01
3427	6.55	18.28	32.29
3476	17.69	76.76	66.79
1458	0.52	9.41	8.99
2259	17.86	34.24	-36.87
2261	-36.89	-21.13	-54.39
2821	83.41	87.79	46.21
2997	-22.65	-40.08	-50.68

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

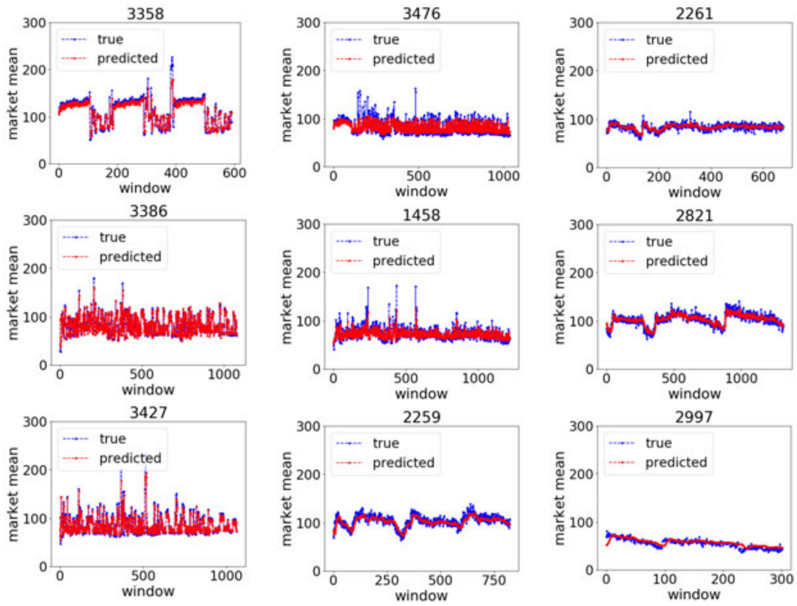

Fig. 2: Market price prediction by LSTM model with window size 500

4.2 Results and Discussions

The prediction of mean market price per window from the ARIMA model is listed in Table 1 with evaluation metric Mean Squared Error (MSE). For comparison, the results of a LSTM model are shown as a percentage of improvements, relative to the results of ARIMA model in Table 2. Positive number means the percentage of the decrease of MSE and vice versa. We take market prices from 10 previous windows as a sequence to predict the price for the next window. Using longer history, like 20 or 50 windows, does not improve the results any further, which suggests short dependency in the temporal space.

Our results show that in most cases, the LSTM model improves the prediction results significantly. It demonstrates the benefits of memorizing more time steps for the prediction. However, there are also performance drops at different aggregation levels in a few campaigns, e.g. 2821 and 2997. We further checked the number of auctions per second for these campaigns. It shows that the pattern of request arrival rate increases or decreases dramatically from training set to test set. In our settings, the window size is fixed. Therefore, with larger window size, the time span in one window varies in this case. The time series model learned in training set fails to capture the changed pattern in the test set. For the other campaigns, the amount of bid requests per second over time remains relatively stable. Given our observations, there is a need to predict the traffic pattern as well, which we did not address in this work. Figure 2 shows the true average market price per window (blue) and the predicted value (red).

Furthermore, we demonstrate how the features in the bid requests contribute to the market price prediction as described in sec.3.3. Table 3 compares the MSE between the results of LSTM taking only historical market price as input and the LSTM model with features in the bid request, LSTM (X). The results clearly show improvements with additional features. However, campaigns 2261 and 2997 have negative results, meaning including additional features increased prediction error. As shown in Figure 2, the mean market price per window in these two campaigns shows very little fluctuation, which suggests a relatively steady environment. In this case, adding additional information from the bid request may perturb the model and jeopardize prediction accuracy. On the contrary, for the campaigns with high volatility, the summary of the requests manages to capture the sudden change in the market.

5 Conclusions and Future work

We provide a temporal market price prediction model for RTB system. Because the dynamics of the market are due to heterogeneous factors such as unknown bidders and different user profiles, we further demonstrate how the information from aggregated bid requests contribute to predicting the volatility of the market. Measuring the uncertainty of the predictions and integrating the price prediction into bidding will be included in the future work.

References

[1] Y. Tagami, S. Ono, K. Yamamoto, K. Tsukamoto, and A. Tajima. Ctr prediction for contextual advertising: Learning-to-rank approach. In *the Seventh International Workshop on Data Mining for Online Advertising*. ACM, 2013.

[2] J. Zhao, G. Qiu, Z. Guan, W. Zhao, and X. He. Deep reinforcement learning for sponsored search real-time bidding. *arXiv preprint arXiv:1803.00259*, 2018.

[3] V. Krishna. *Auction theory*. Academic press, 2009.

[4] J. Reisinger and M. Driscoll. Pricing externalities in real-time bidding markets. In *NIPS Workshop: Machine Learning in Online Advertising*, 2010.

[5] W. Zhang, S. Yuan, and J. Wang. Optimal real-time bidding for display advertising. In *KDD*. ACM, 2014.

[6] Y. Yuan, F. Wang, J. Li, and R. Qin. A survey on real time bidding advertising. In *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics*, pages 418–423. IEEE, 2014.

[7] M. Du, R. Sassioui, G. Varisteas, M. Brorsson, O. Cherkaoui, and R. State. Improving real-time bidding using a constrained markov decision process. In *International Conference on Advanced Data Mining and Applications*. Springer, 2017.

[8] 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 *WSDM*. ACM, 2017.

[9] W. Zhang, T. Zhou, J. Wang, and J. Xu. Bid-aware gradient descent for unbiased learning with censored data in display advertising. In *KDD*. ACM, 2016.

[10] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8), 1997.

[12] Weinan Zhang, Shuai Yuan, and Jun Wang. Real-time bidding benchmarking with ipinyou dataset. *CoRR*, abs/1407.7073, 2014.