Functional Bid Landscape Forecasting for Display Advertising

Yuchen Wang[1], Kan Ren[1], Weinan Zhang[1(✉)], Jun Wang[2], and Yong Yu[1(✉)]

[1] Shanghai Jiao Tong University, Shanghai, China
{zeromike,kren,wnzhang,yyu}@apex.sjtu.edu.cn
[2] University College London, London, UK
j.wang@cs.ucl.ac.uk

Abstract. Real-time auction has become an important online advertising trading mechanism. A crucial issue for advertisers is to model the market competition, i.e., *bid landscape forecasting*. It is formulated as predicting the market price distribution for each ad auction provided by its side information. Existing solutions mainly focus on parameterized heuristic forms of the market price distribution and learn the parameters to fit the data. In this paper, we present a functional bid landscape forecasting method to automatically learn the function mapping from each ad auction features to the market price distribution without any assumption about the functional form. Specifically, to deal with the categorical feature input, we propose a novel decision tree model with a node splitting scheme by attribute value clustering. Furthermore, to deal with the problem of right-censored market price observations, we propose to incorporate a survival model into tree learning and prediction, which largely reduces the model bias. The experiments on real-world data demonstrate that our models achieve substantial performance gains over previous work in various metrics. The software related to this paper is available at https://github.com/zeromike/bid-lands.

1 Introduction

Popularized from 2011, real-time bidding (RTB) has become one of the most important media buying mechanism in display advertising [7]. In RTB, each ad display opportunity, i.e., an ad impression, is traded through a real-time auction, where each advertiser submits a bid price based on the impression features and the one with the highest bid wins the auction and display her ad to the user [20]. Apparently, the bidding strategy that determines how much to bid for each specific ad impression is a core component in RTB display advertising [16].

As pointed out in [22], the two key factors determining the optimal bid price in a specific ad auction are *utility* and *cost*. The utility factor measures the value of ad impression, normally quantified as user's response rate of the displayed ad, such as click-through rate (CTR) or conversion rate (CVR) [12]. The cost factor, on the other hand, estimates how much the advertiser would need to pay to win the ad auction [3]. From an advertiser's perspective, the *market price* is defined

P. Frasconi et al. (Eds.): ECML PKDD 2016, Part I, LNAI 9851, pp. 115–131, 2016.
DOI: 10.1007/978-3-319-46128-1_8

as the highest bid price from her competitors[1]. In the widely used second-price auctions, the winner needs to pay the second highest bid price in the auction, i.e., the market price [4]. Market price estimation is a difficult problem because it is the highest bid from hundreds or even thousands of advertisers for a specific ad impression, which is highly dynamic and it is almost impossible to predict it by modeling each advertiser's strategy [2]. Thus, the practical solution is to model the market price as a stochastic variable and to predict its distribution given each ad impression, named as *bid landscape*.

Previous work on bid landscape modeling is normally based on predefining a parameterized distribution form, such as Gaussian distribution [18] or log-normal distribution [3]. However, as pointed out in [19], such assumptions are too strong and often rejected by statistical tests. Another practical problem is the observed market price is right-censored, i.e., only when the advertiser wins the auction, she can observe the market price (by checking the auction cost), and when she loses, she only knows the underlying market price is higher than her bid. Such censored observations directly lead to biased landscape models.

In this paper, we present a novel *functional bid landscape forecasting* model to address the two problems. Decision tree is a method commonly used in data mining [5,17]. By building a decision tree, the function mapping from the auctioned ad impression features to the corresponding market price distribution is automatically learned, without any functional assumption or restriction. More specifically, to deal with the categorical features which are quite common in online advertising tasks, we propose a novel node splitting scheme by performing clustering based on the attribute values, e.g., clustering and splitting the cities. The learning criterion of the tree model is based on KL-Divergence [10] between the market price distributions of children nodes. Furthermore, to model the censored market price distribution of each leaf node, we adopt non-parametric survival models [9] to significantly reduce the modeling bias by leveraging the lost bid information.

The experiments on a 9-advertiser dataset demonstrate that our proposed solution with automatic tree learning and survival modeling leads to a 30.7 % improvement on data log-likelihood and a 77.6 % drop on KL-Divergence compared to the state-of-the-art model [18].

In sum, the technical contributions of this paper are three-fold.

Automatic function learning: a decision tree model is proposed to automatically learn the function mapping from the input ad impression features to the market price distribution, without any functional assumption.

Node splitting via clustering: the node splitting scheme of the proposed tree model is based on the KL-Divergence maximization between the split data via a K-means clustering of attribute values, which naturally bypasses the scalability problem of tree models working on categorical data.

[1] The terms 'market price' and 'winning (bid) price' are used interchangeably in related literature [2,3,18]. In this paper, we use 'market price'.

Efficient censorship handling: with a non-parametric survival model, both the data of observed market prices and lost bid prices are fed into the decision tree learning to reduce the model bias caused by the censored market price observations.

The rest of this paper is organized as follows. We discuss some related work and compare with ours in Sect. 2. Then we propose our solution in Sect. 3. The experimental results and detailed discussions are provided in Sect. 4. We finally conclude this paper and discuss the future work in Sect. 5.

2 Related Work

Bid Landscape Forecasting. As is discussed above, bid landscape forecasting is a crucial component in online advertising framework, however, lacking enough attention. On one hand, researchers proposed several heuristic forms of functions to model the market price distribution. In [22], the authors provided two forms of winning probability w.r.t. the bid price, which is based on the observation of an offline dataset. However, this derivation has many drawbacks since the appropriate distribution of the market price in real world data may deviate much from the simple functional form. On the other hand, some fine-studied distributions are also used in market price modeling. [3] proposed a log-normal distribution to fit the market price distribution. The main drawback is that these distributional methods may lose the effectiveness of handling various dynamic data and they ignore the real data divergency as we will show later in Figs. 1 and 3.

In view of forecasting, [3] presented a template-based method to fetch the corresponding market price distribution w.r.t. the given auction request. However, this paper studied the problem as on the seller side which is quite different from the buyer side as we stand. [18] proposed a regression method to model the market price w.r.t. auction features. However, those methods do not care much about the real data properties, i.e. similarity and distinction among data segments, which may result in poor forecasting performance on different campaigns. Moreover, none of the above methods deal with the data censorship problem in modeling training.

Bid Optimization. Bid optimization is a well studied problem and has drawn many concerns both in RTB environment [1,21,22]. This task aims to optimize the strategies to allocate budget to gain ad display opportunities [1], so it is crucial to model the market competition and make accurate bid landscape prediction [3]. In RTB display advertising with cost-per-impression scheme, the bid decision is made on the level of impression so that the price is also charged on impression level [11]. It again emphasizes key importance of the forecasting task in online bidding. In [21,22], the authors proposed a functional optimization method with the consideration of budget constraints and market price distribution, which led to an optimal bidding strategy. However, the market price

distribution adopted in these two papers is under heuristic assumptions, which may not perform well in real-world forecasting tasks.

Learning over Censored Data. In machine learning fields, dealing with censored data is sometimes regarded as handling missing data, which is a well-studied problem [6]. The item recommendation task with implicit feedback is a classic problem of dealing with missing data. [15] proposed a uniform sampling of negative feedback items for user's positive ones [14]. In the online advertising field, the authors in [18] proposed a regression model with censored regression module using the lost auction data to fix the biased data problem. However, the Gaussian conditional distribution assumption turns out to be too strong, which results in weak performance in our experiment. The authors in [2] implemented a product-limit estimator [9] in handling the data censorship in sponsored search, but the bid landscape is built on search keyword level, which is not fine-grained to work on RTB display advertising. We transfer the survival analysis method from [2] to RTB environment and compare with [18] in our experiment.

3 Methodology

3.1 Problem Definition

The goal of bid landscape forecasting is to predict the probabilistic distribution density (p.d.f.) $p_{\boldsymbol{x}}(z)$ w.r.t. the market price z given an ad auction information represented by a high-dimensional feature vector $\boldsymbol{x}$.

Table 1. The statistics of attributes.

Attribute	Adex-change	Weekday	Slot-visibility	Slotheight	Slotwidth	Hour	Region	User-agent	Creative	City
Num of values	5	7	11	14	21	24	35	40	131	370

Each auction $\boldsymbol{x}$ contains multiple side information, e.g. user agent, region, city, user tags, ad slot information, etc. In Table 1, we present the attributes contained in the dataset with corresponding numbers of value. We can easily find that different attributes vary in both diversity and quantity. Moreover, the bid price distribution of a given request may be diverse in different attributes. Take the field `Region` as an example, the bid distribution of the samples with region in `Beijing` is quite different from that of `Xizang`, which is illustrated in Fig. 1. Previous work focuses only on the heuristic forms (e.g. log-normal [3] or a unary function [22]) of distribution and cannot effectively capture the divergency within data.

Moreover, in RTB marketplace, the advertiser proposes a bid price b and wins if $b > z$ paying z for the ad impression, loses if $b \leq z$ without knowing the exact value of z, where z represents the market price which is the highest bid price

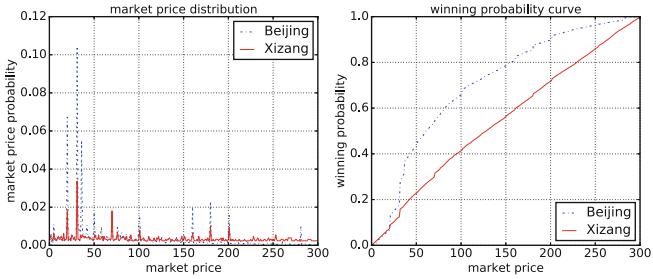

Fig. 1. Market price distribution over different regions.

from the competitors. Apparently, the true market price is only observable for who is winning the corresponding auction. As for the lost auctions, the advertiser only knows the lower bound of the market price, which results in the problem of right censored data [2]. The censorship from the lost auctions may heavily influence the forecasting performance in the online prediction [18].

In this paper, we mainly settle down these two problems. First, we propose to automatically build the function mapping from the given ad auction $\boldsymbol{x}$ to the market price distribution $p_{\boldsymbol{x}}(z)$ without any functional form assumption, generally represented as

$$p_{\boldsymbol{x}}(z) = T_p(\boldsymbol{x}). \tag{1}$$

Second, we leverage both the observed market price data of winning auctions and censored one of losing auctions to train a less biased function $T_p(\boldsymbol{x})$.

We use a binary decision tree to represent $T_p(\boldsymbol{x})$. More precisely, every node represents a set of auction samples. For each node O_i, we split the contained samples into two sets $\{S_{ij}^t\}$ according to attribute A_j (e.g. `Region`) value sets (e.g. {`Xizang`, `Beijing`, ...}), where $t \in \{1, 2\}$, $A_j \in \Theta$ and Θ is the attribute space. For each subset S_{ij}^t, the corresponding market price distribution $p_{\boldsymbol{x}}^t(z)$ can be statistically built. Intuitively, different subsets have diverse distributions and the samples within the same subset are similar to each other, which requires effective clustering and node splitting scheme. Furthermore, KL-Divergence [10] is a reasonable metric to measure the splitted data divergency. So that we choose the best splitting π_i with the highest KL-Divergence value D_{KL}^i calculated between the resulted two subsets $S_{i\cdot}^1$ and $S_{i\cdot}^2$ in node O_i. Essentially, our goal is to seek the splitting strategy $\pi = \cup_{i \in I} \pi_i$, where each splitting action π_i maximizes the KL-Divergence D_{KL} between two child sets in node O_i. Mathematically, our functional bid landscape forecasting system is built as

$$T_p^{\pi}(\boldsymbol{x}) = \arg\max_{\pi} \sum_{i=1}^{l} D_{\mathrm{KL}}^{i}, \tag{2}$$

$$D_{\mathrm{KL}}^{i} = \max\{D_{\mathrm{KL}}^{i1}, D_{\mathrm{KL}}^{i2}, ..., D_{\mathrm{KL}}^{ij}, ..., D_{\mathrm{KL}}^{iN}\} \tag{3}$$

$$D_{\mathrm{KL}}^{ij} = \sum_{z=1}^{z_{\max}} p_{\boldsymbol{x}}(z) \log \frac{p_{\boldsymbol{x}}(z)}{q_{\boldsymbol{x}}(z)}, \tag{4}$$

where p and q are the two probability distributions for the splitted subsets, $z_{\max}$ represents the maximum market price, D_{KL}^{ij} means the maximum KL-Divergence of splitting over the sample set of attribute A_j in node i, $N = |\Theta|$ is the number of attributes and l is the number of splitting nodes.

When forecasting, every auction instance will follow a path from the root to the leaf, classified by the dividing strategy according to the attribute value it contains. The bid landscape $p_{\boldsymbol{x}}(z)$ is finally predicted at the leaf node.

3.2 Decision Trees with K-means Clustering

In this section, we propose the K-means clustering method based on KL-Divergence. Then, we will present our iterative optimization algorithm for the decision tree learning model with K-means clustering.

K-Means Clustering with KL-Divergence. For each attribute, the values vary over different samples, as we can see in Table 1. The goal of the binary decision tree spanning is to group similar values w.r.t. one attribute and split the data samples into two divergent subsets. We use KL-Divergence to model the statistics of datasets, which is shown in Eq. (2). KL-Divergence is a measurement assessing the difference between two probability distributions. The problem need to solve is that it requires an effective clustering method to group samples w.r.t. the given metric.

In this paper, the bid samples with the same attribute value are considered as a single point. The goal of K-means clustering is to partition these points into two clusters according to the calculated KL-Divergence. The process of K-means clustering summarize in Algorithm 1.

The input of Algorithm 1 is the attribute A_j and a set of training samples $S = \{s_1, s_2, ..., s_k, ..., s_n\}$, where s_k is the set of training samples with the same value for attribute A_j and n is the number of different values of attribute A_j.

We adopted an iterative algorithm to achieve the clustering goal. First, we randomly split the data into two parts S^1 and S^2. Then we propose an EM-fashion algorithm to iterate two steps below until the whole process converges to the optimal objective, i.e., maximize the KL-Divergence.

E-step: Compute the market price probabilistic distribution Q_1 for S^1 and Q_2 for S^2, which will be discussed in Sect. 3.3.

M-step: Consider the sample data with the same value of attribute A_j as a whole, we will get $\{s_1, s_2, ..., s_k, ..., s_n\}$ if the attribute A_j has n different values. And we will have n corresponding market price probability distributions

Algorithm 1. K-Means clustering with KL-Divergence

Input: Training sample $S = \{s_1, s_2, ..., s_n\}$; Attribute A_j;
Output: KL-Divergence D^j_{KL} over attribute A_j, Data clusters S^1 and S^2;

```
Randomly split the data into two parts S^1 and S^2;
while not converged do
    E-step:
    Get price probability distribution Q_1 for S^1 and Q_2 for S^2;
    M-step:
    for all M_k,k ∈ {1, 2, 3, ..., n} do
        Calculate the K_1 between M_k and Q_1 by Eq. (4);
        Calculate the K_2 between M_k and Q_2 by Eq. (4);
        Update S^1 or S^2 by comparing with K_1 and K_2;
    end for
    Calculate the D^j_KL between Q_1 and Q_2 by Eq. (4);
end while
Return D^j_KL, S^1 and S^2;
```

$\{M_1, M_2, ..., M_n\}$. For each market price probability distribution M_k, we will calculate the KL-Divergence K_1 between M_k and Q_1, K_2 between M_k and Q_2, respectively. If $K_1 > K_2$, it means that the probability distribution M_k is more similar with Q_2, thus assign data set s_k to the relatively more similar data set S^2, vice versa.

After each M-step, we will calculate the KL-Divergence D^j_{KL} between Q_1 and Q_2. The EM iteration stops when D^j_{KL} does not change.

In this paper, we only split each node into two subsets, i.e., $k = 2$. To avoid bringing another control variable into the model, we do not discuss about cases of $k >= 3$. As a result, we choose $k = 2$ to make it consistent with the bi-spliting scheme on numeric features.

Building the Decision Tree. The combined scheme of building decision tree based on K-means clustering node splitting is described in Algorithm 2. In Algorithm 2, we first find the splitting attribute with highest KL-Divergence. Then, we perform the binary splitting of the data by maximizing KL-Divergence between two leaf nodes with K-means clustering. The sub-tree keeps growing until the length of sample data in leaf node is less than a predefined value. Finally, we prune the tree by using reduced error pruning method. Compared with the decision tree algorithm, the main difference of our proposed scheme is that the binary node splitting scheme with K-means clustering and the usage of KL-Divergence as the attribute selection criteria.

In the test process, one problem is that there could be new attribute values of some test data instances which do not match any nodes of our decision tree learned from the training data. To handle this, we deploy a randomly choosing method which decides the attribute value of the given test data to randomly goes to one of the two children, which is equivalent to non-splitting on such

Algorithm 2. Building Decision Tree with K-Means clustering

Input: Training sample S which contain N attributes;

1: **for** all attribute A_j,$j \in \{1, 2, 3, ..., N\}$ **do**
2: Calculate the KL-Divergence D_{KL}^{j} for attribute A_j by Algorithm 1;
3: **end for**
4: $D_{\mathrm{KL}}^{\mathrm{best}} = \max\{D_{\mathrm{KL}}^{1}, D_{\mathrm{KL}}^{2}, ..., D_{\mathrm{KL}}^{j}, D_{\mathrm{KL}}^{N}\}$;
5: Find A_{best} with $D_{\mathrm{KL}}^{\mathrm{best}}$;
6: Create a decision node that splits on A_{best};
7: Split the decision node into two nodes S^1 and S^2;
8: Return new nodes as children of the parent node

attribute. The experiment results show that such random method works well on the real-world dataset.

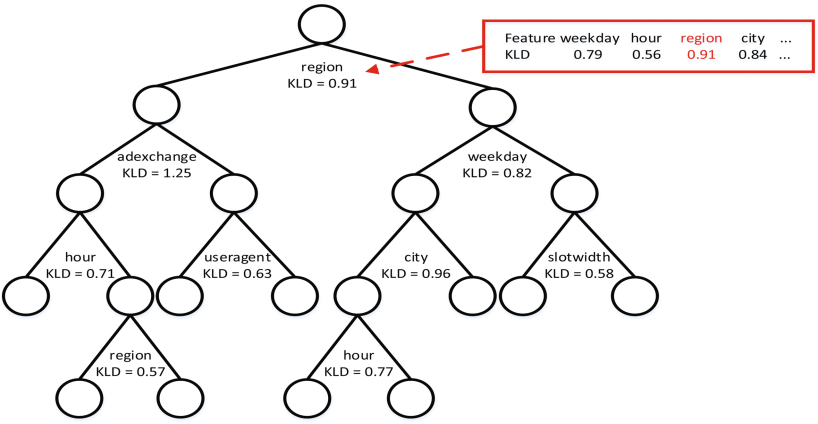

Fig. 2. Illustration of the tree.

Figure 2 is an example of the decision tree. As we can see, for each node, we illustrate its best splitting attribute and the corresponding KL-Divergence. The red box shows the KL-Divergence value for each attribute, and the best splitting attribute with the highest KL-Divergence is chosen.

3.3 Bid Landscape Forecasting with Censored Data

In real-time bidding, an advertiser only observes the market prices of the auctions that she wins. For those lost auctions, she only knows the lower bound of the market price, i.e., her bid price. Such data is named as the right censored data [18]. However, the partial information in lost auctions is still of high value. To better estimate the bid distribution, we introduce survival models [8] to model the censored data. We implement a non-parametric method to model the real

market price distribution and transfer survival analysis from keywork search advertising [2] to RTB environment. That is, given the observed impressions and the lost bid requests, the winning probability can be estimated with the non-parametric Kaplan-Meier Product-Limit method [9].

Suppose we have sequential bidding logs in form of $\{b_i, w_i, m_i\}_{i=1,2,\ldots,M}$, where b_i is the bidding price in the auction, w_i is the boolean value of whether we have won the auction or not, and m_i is the market price (unknown if $w_i = 0$). Then we transform our data into the form of $\{b_j, d_j, n_j\}_{j=1,2,\ldots,N}$, where the bidding price $b_j < b_{j+1}$, and d_j represents the number of the winning auctions with bidding price $b_j - 1$, n_j is the number of auctions that cannot not be won with bidding price $b_j - 1$. Then the probability of losing an auction with bidding price b_x is

$$l(b_x) = \prod_{b_j < b_x} \frac{n_j - d_j}{n_j}. \tag{5}$$

Thus the winning probability $w(b_x)$ and the integer[2] market price p.d.f. $p(z)$ are

$$w(b_x) = 1 - \prod_{b_j < b_x} \frac{n_j - d_j}{n_j}, \quad p(z) = w(z+1) - w(z). \tag{6}$$

4 Experiments

In this section, we introduce the experimental setup and analyze the results[3]. We compare the overall performance over 5 different bid landscape forecasting models, and further analyze the performance of our proposed against different hyperparameters (e.g. tree depth, leaf size).

4.1 Dataset

For the following experiments, we use the real-world bidding log from iPinYou RTB dataset[4]. It contains 64.7 M bidding records, 19.5 M impressions, 14.79 K clicks and 16.0 K CNY expense on 9 campaigns from different advertisers during 10 days in 2013. Each bidding log has 26 attributes, including weekday, hour, user agent, region, slot ID etc. More details of the data is provided in [13].

4.2 Experiment Flow

In order to simulate the real bidding market and show the advantages of our survival model, we take the original data of impression log as full-volume auction data, and perform a truthful bidding strategy [12] to simulate the bidding process, which produces the winning bid dataset W and lost bid dataset L respectively. For each data sample $\boldsymbol{x}_{\text{win}} \in W$, the simulated real market price z_{win} is

[2] In practice, the bid prices in various RTB ad auctions are required to be integer.
[3] The experiment code is available at http://goo.gl/h130Z0.
[4] Dataset link: http://data.computational-advertising.org.

known for the advertisers, while the corresponding market price z_{lose} remaining unknown for $\boldsymbol{x}_{\text{lose}} \in L$. It guarantees the similar situation as that faced by all the advertisers in the real world marketplace.

In the test phase, the corresponding market price distribution $p_{\boldsymbol{x}}(z)$ of each sample $\boldsymbol{x}$ in the test data is estimated by all of the compared models respectively. We assess the performance of different settings in several measurements, as listed in the next subsection. Finally we study the performance of our proposed model with different hyperparameters, e.g., the tree depth and the maximum size of each leaf.

4.3 Evaluation Measures

The goal of this paper is to improve the performance of market price distribution forecasting. We use two evaluation methods to measure the forecasting error. The first one is Average Negative Log Probability (ANLP). After we classifying each sample data into different leaves with the tree model, the sum of log probability for all sample data P_{nl} is given by the Eq. (7), and the average negative log probability $\bar{P}_{\text{nl}}$ given by the $\bar{P}_{\text{nl}}$:

$$P_{\text{nl}} = \sum_{i=1}^{k} \sum_{j=1}^{z_{\max}} (-\log P_{ij}) N_{ij}, \tag{7}$$

$$N = \sum_{i=1}^{k} \sum_{j=1}^{z_{\max}} N_{ij}, \quad \bar{P}_{\text{nl}} = P_{\text{nl}}/N, \tag{8}$$

where k denotes the number of sub bid landscapes, $z_{\max}$ represents the maximum market price, P_{ij} means the probability of training sample in the ith leaf node given price j, N_{ij} is the number of test sample in the ith leaf node given price j. N is the total number of test samples.

We also calculate the overall KL-Divergence to measure the objective forecasting error. D_{KL} is given by the Eq. (9):

$$D_{\text{KL}} = \frac{1}{N} \sum_{i=1}^{k} N_i \sum_{j=1}^{z_{\max}} P_{ij} \log \frac{P_{ij}}{Q_{ij}}, \tag{9}$$

where N_i means the number of test sample in the ith leaf node. Q_{ij} means the probability of test sample in the ith leaf node given price j.

4.4 Compared Settings

We compare five different bid landscape forecasting models in our experiment.

NM - The Normal Model predicts the bid landscape based on the observed market prices from simulated impression log W, without using the lost bid request data in L. This model uses a non-parametric method to directly draw the probability function w.r.t. the market price from the winning dataset.

SM - The Survival Model forecasts the bid landscape with survival analysis, which learns from both observed market prices from impression log and the lost bid request data using Kaplan-Meier estimation [2]. The detail has been discussed in Sect. 3.3.

MM - The Mix Model uses linear regression and censored regression to predict the bid landscape respectively, and combines two models considering winning probability into Mixture Model [18] to predict the final bid landscape.

NTM - The Normal Tree Model predicts the bid landscape using only our proposed tree model, without survival analysis. The detailed modeling method has been declared in Sect. 3.2.

STM - The Survival Tree Model predicts the bid landscape with the proposed survival analysis embedded in our tree model, which is our final mixed model.

4.5 Experiment Results

Data Analysis. Table 2 shows the overall statistics of the dataset, where each row presents the statistical information of the corresponding advertiser in the first column. In Table 2, *Num of bids* is the number of total bids, and *Num of win bids* is the number of winning bids in the full simulated dataset $W \cup L$. *WR* is the winning rate calculated by $\frac{|W|}{|W \cup L|}$. *AMP* is the average market price on all bids. *AMP on W* and *AMP on L* are the average market price for the winning bid set W and the lost bid set L, respectively.

We can easily find that the winning rates of all campaigns are low, which is practically reasonable since a real-world advertiser can only win a little proportion of the whole-world volume. The market prices of most impressions are unavailable to the advertiser. We also observe that the average market price on winning bids (*AMP on W*) are much lower than average market price on lost bids (*AMP on L*). This verifies the bias between the observed market price distribution and the true market price distribution.

Table 2. The statistics of the dataset iPinYou.

Advertiser	Num of bids	Num of win bids	WR	AMP	AMP on W	AMP on L
1458	2,055,371	257,077	0.1251	70.2829	29.1130	76.1684
2259	557,038	135,487	0.2432	96.0685	28.3345	117.8383
2261	458,412	176,325	0.3846	92.1654	32.3218	129.5721
2821	881,708	305,134	0.3461	90.6573	35.0455	120.0881
2997	208,292	60,556	0.2907	64.9918	16.6269	84.8163
3358	1,161,403	336,769	0.2900	92.6624	55.6009	107.7978
3386	1,898,535	332,223	0.1750	80.4224	37.4947	89.5276
3427	1,729,177	563,592	0.3259	82.6685	53.1614	96.9359
3476	1,313,574	303,341	0.2309	79.4990	38.7343	91.7393
Overall	10,263,510	2,470,504	0.2407	81.9769	41.1313	94.9256

Bid Landscapes of Leaf Nodes. There are 4 examples of bid landscape between training and testing samples shown on Fig. 3. From the figures, we can find that the bid landscape of each leaf node is quite different from that of other leaf nodes. Especially, some sub bid landscape tends to have a large probability of some price, and the training distribution fit the test distribution very well. This result suggests we can predict the bid landscape more accurately with tree models.

Survival Model. As is mentioned in Sect. 3.3, the observed market price distribution is biased due to the data censorship. Figure 4 shows the comparison of the curves for market price distribution and winning probability. TRUTH represents for the real market price distribution for the test data, which is regarded as the ground truth. FULL curve is built from full-volume data, i.e., assume the advertiser has observed all market prices of $W \cup L$, which is regarded as the upperbound performance of any bid landscape model based on censored data. We observe that (i) FULL curve is the most close to TRUTH since FULL makes use of full-volume training data and is naturally unbiased. However, in the practice, advertisers only have a small number of winning logs W [13]. (ii) Compared to NM, STM curve is much more close to TRUTH, which verifies its advantage of making use of the censored data with survival analysis to improve the performance of market price distribution forecasting.

Performance Comparison. We evaluate on five models described in Sect. 4.4 with evaluation measure given in Sect. 4.3. Table 3 presents the Average negative log probability (ANLP) and KL-Divergence (KLD) of these settings.

For ANLP, we observe that (i) for all campaigns investigated, STM shows the best performance, which verifies the effectiveness of the survival tree model. (ii) SM is better than NM because SM learns from both winning bids and lost bids to handle the censored data problem. (iii) We shall notice that NTM is the tree version of NM, and STM is the tree version of SM. We find that NTM outperforms NM, and STM outperforms SM, which means the tree model effectively improves the performance of bid landscape forecasting. (iv) STM is the combination of SM and NTM, both of which contribute to a better performance as is mentioned in (ii) and (iii). Thus it is reasonable that STM has the best performance. It has both advantages of SM and NTM, i.e., dealing with the bid

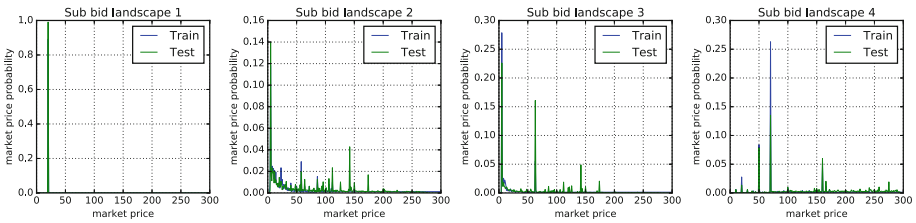

Fig. 3. Examples of different sub bid landscapes.

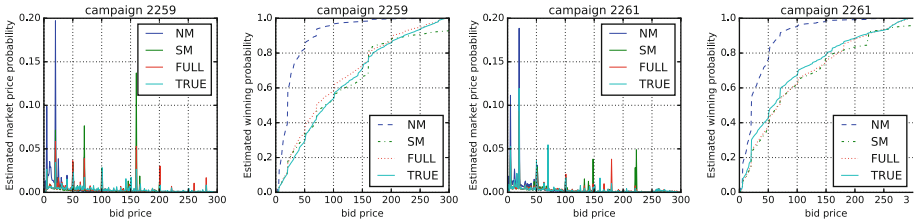

Fig. 4. Comparison of the curves of market price distribution and winning probability.

Table 3. Performance illustration. Average negative probability of five compared settings. ANLP: the smaller, the better. KLD: the smaller, the better.

Campaign	ANLP					KLD				
	MM	NM	SM	NTM	STM	MM	NM	SM	NTM	STM
1458	5.7887	5.3662	4.7885	4.7160	4.3308	0.7323	0.7463	0.2367	0.6591	0.2095
2259	7.3285	6.7686	5.8204	5.4943	5.4021	0.8264	0.9633	0.3709	0.8757	0.1668
2261	7.0205	5.5310	5.1053	4.4444	4.3137	1.0181	0.4029	0.2943	0.3165	0.1222
2821	7.2628	6.5508	5.6710	5.4196	5.3721	0.7816	0.9671	0.3562	0.6170	0.2880
2997	6.7024	5.3642	5.1411	5.1626	5.0944	0.7450	0.4526	0.1399	0.3312	0.1214
3358	7.1779	5.8345	5.2771	4.8377	4.6168	1.4968	0.8367	0.5148	0.8367	0.3900
3386	6.1418	5.2791	4.8721	4.6698	4.2577	0.8761	0.6811	0.3474	0.6064	0.2236
3427	6.1852	4.8838	4.6453	4.1047	4.0580	1.0564	0.3247	0.1478	0.3247	0.1478
3476	6.0220	5.2884	4.7535	4.3516	4.2951	0.9821	0.6134	0.2239	0.5650	0.2238
Overall	6.5520	5.6635	5.0997	4.7792	**4.6065**	0.9239	0.6898	0.2927	0.5834	**0.2160**

distribution difference between different attribute value and learning from the censored data.

For KLD, we can also find that STM achieves the best performance. The results of other models are also similar to those of ANLP, but there are some interesting differences. Note that for campaign 3427, the KL-Divergence values of NM and NTM are equal to each other, so do SM and STM. The KL-Divergence values of SM and STM for Campaign 3476 are also nearly the same. That is because the optimal depth of tree in these cases is 1. We shall notice that actually NM and SM are the special cases of NTM and STM respectively when the tree depthes of the latter two models are equal to 1. The fact arouses the question, how to decides the optimal tree depth? We here take the tree depth as a hyperparameter, and we leave the detailed discussion in the next subsection.

Table 4. p-values in t-tests of ANLP comparison.

Model	MM	NM	SM	NTM
STM	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$

As is mentioned above, in terms of KLD, SM and STM for campaign 3476 are actually the same model since the optimal tree depth of STM for campaign 3476 is 1. One may still find that the KLD of SM and STM for campaign 3476 is a little different. That is caused by the handling method of missing feature values in training data, which is described in Sect. 3.2. As the experiment result shows, the influence is negligible.

We deploy a *t-test* experiment on negative log probability between our proposed model STM and each of other compared settings to check the statistical significance of the improvement. Table 4 shows that the p-value of each test is lower than 10^{-6}, which means the improvement is statistically significant. The significant test on KL-Divergence is not performed because KLD is not a metric calculated based on each data instance.

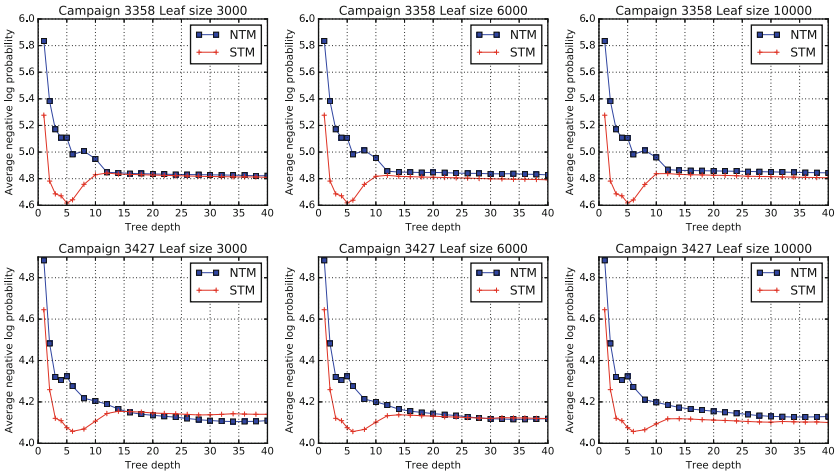

Fig. 5. Average negative log probability with different tree depth.

Hyperparameter Tuning. There are two problems in decision tree algorithms. If we do not limit the size of tree, it will split into many quite small sets, and overfit the training and fail to generalize on new data. However, if the limitation for the size of tree is too much, some nodes that have useful information cannot be split in succession, which is known as horizon effect. In order to avoid both problems, we need to find out the optimal size limitation of the tree. There are two hyperparameters that influence the size of tree, i.e., (i) tree depth (the upperbound of tree depth) and (ii) leaf size (the upperbound of number of training samples in a leaf). In the experiment, we kept changing these two hyperparameters, and compare the average negative log probability with different values. The results are illustrated in Fig. 5. Different pictures represent different limitation on leaf size.

We observe that, for most campaigns, (i) the performance of NTM improves finally converges as the tree depth grows. (ii) The performance of STM improves

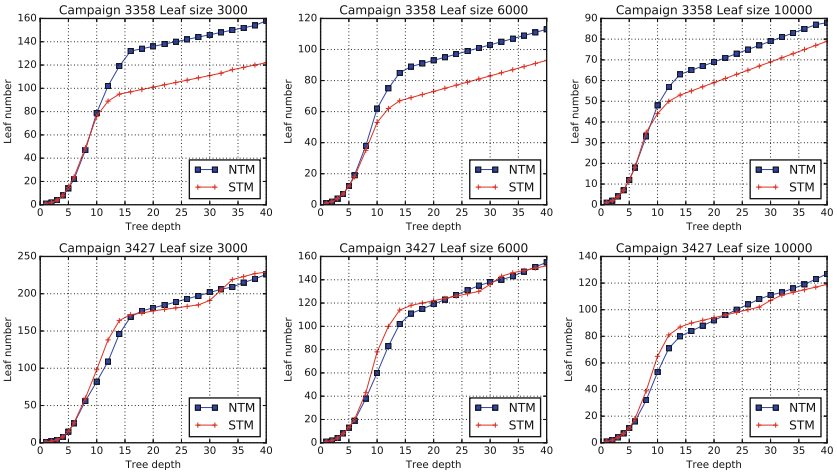

Fig. 6. Relationship between the leaf number and the tree depth.

at first as the tree depth grows, but after tree depth exceeds a certain value, the performance is getting worse, which can be explained as overfitting. (iii) When the tree depth is large enough, the effect of survival model is weakened. The performance of STM and NTM in this case is almost the same. (iv) The leaf size will affect the performance of the tree model, but the overall influence mainly occurs when the tree depth is large. Since the optimal depth for NTM is usually large, the leaf size tends to have a larger influence on performance of NTM. While the optimal depth for STM is usually small, the leaf size will have a smaller influence on STM's performance.

Figure 6 shows the relationship between the leaf number and the tree depth. We can find that the number of leaf increases rapidly at first. When the depth of tree grows up, the growth of leaf number begins to slow down, which corresponds to the convergence of ANLP shown in Fig. 5.

Table 5. The average optimal tree depth and leaf numbers for different models.

	Tree depth		Leaf number	
Model	ANLP	KLD	ANLP	KLD
NTM	20.33	11.33	632.67	398.67
STM	5.89	4.89	25.33	52.11

In the experiment, we use a validation set to find out the optimal tree depth. Table 5 shows the average optimal tree depths and the corresponding leaf numbers for NTM and STM. We can find that the average optimal tree depth and leaf number of STM is lower than that of NTM, because STM learns from both

winning bids and lost bids. As the tree grows, it will reach the best performance earlier than NTM, which only learns from the winning bids.

We also experimentally illustrate the EM convergence of the tree model in Fig. 7, which shows the value changes of KL-Divergence over EM training rounds. We observe that our optimization converges within about 6 EM rounds, and the fluctuation is small. In our experiments, the EM algorithm is quite efficient and converges quickly. The average training rounds of our EM algorithm is about 4.

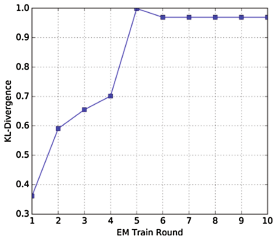

Fig. 7. KL-Divergence convergence w.r.t. EM rounds.

5 Conclusion and Future Work

In this paper, we have proposed a functional bid landscape forecasting methodology in RTB display advertising, which automatically builds a function mapping from the impression features to the market price distribution. The iterative learning framework trains a decision tree by clustering-based node splitting with the KL-Divergence objective. We also incorporate the survival model to handle the model bias problem caused by the censored data observations. The overall model significantly improves the forecasting performance over the baselines and the state-of-the-art models in various metrics.

In the future work, we plan to combine the functional bid landscape forecasting with utility (e.g. click-through rate, conversion rate) estimation model, aiming to make more reasonable and informative decisions in bidding strategy.

References

1. Agarwal, D., Ghosh, S., Wei, K., You, S.: Budget pacing for targeted online advertisements at linkedin. In: KDD (2014)
2. Amin, K., Kearns, M., Key, P., Schwaighofer, A.: Budget optimization for sponsored search: Censored learning in mdps. arXiv preprint arXiv:1210.4847 (2012)
3. Cui, Y., Zhang, R., Li, W., Mao, J.: Bid landscape forecasting in online ad exchange marketplace. In: KDD (2011)
4. Edelman, B., Ostrovsky, M., Schwarz, M.: Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. Technical report, National Bureau of Economic Research (2005)

5. Faddoul, J.B., Chidlovskii, B., Gilleron, R., Torre, F.: Learning multiple tasks with boosted decision trees. In: Flach, P.A., Bie, T., Cristianini, N. (eds.) ECML PKDD 2012. LNCS (LNAI), pp. 681–696. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33460-3_49
6. García-Laencina, P.J., Sancho-Gómez, J.L., Figueiras-Vidal, A.R.: Pattern classification with missing data: a review. Neural Comput. Appl. **19**(2), 263–282 (2010)
7. Google: The arrival of real-time bidding (2011)
8. Johnson, N.L.: Survival Models and Data Analysis. Wiley, New York (1999)
9. Kaplan, E.L., Meier, P.: Nonparametric estimation from incomplete observations. J. Am. Stat. Assoc. **53**, 457–481 (1958)
10. Kullback, S.: Letter to the editor: the kullback-leibler distance (1987)
11. Lee, K.C., Jalali, A., Dasdan, A.: Real time bid optimization with smooth budget delivery in online advertising. In: ADKDD (2013)
12. Lee, K.c., Orten, B.B., Dasdan, A., Li, W.: Estimating conversion rate in display advertising from past performance data (2012)
13. Liao, H., Peng, L., Liu, Z., Shen, X.: ipinyou global rtb bidding algorithm competition dataset. In: ADKDD (2014)
14. Marlin, B.M., Zemel, R.S.: Collaborative prediction and ranking with non-random missing data. In: RecSys (2009)
15. Pan, R., Zhou, Y., Cao, B., Liu, N.N., Lukose, R., Scholz, M., Yang, Q.: One-class collaborative filtering. In: ICDM (2008)
16. Perlich, C., Dalessandro, B., Hook, R., Stitelman, O., Raeder, T., Provost, F.: Bid optimizing and inventory scoring in targeted online advertising. In: KDD (2012)
17. Stojanova, D., Ceci, M., Appice, A., Džeroski, S.: Network regression with predictive clustering trees. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011. LNCS (LNAI), pp. 333–348. Springer, Heidelberg (2011). doi:10.1007/978-3-642-23808-6_22
18. Wu, W.C.H., Yeh, M.Y., Chen, M.S.: Predicting winning price in real time bidding with censored data. In: KDD (2015)
19. Yuan, S., Wang, J., Chen, B., Mason, P., Seljan, S.: An empirical study of reserve price optimisation in real-time bidding. In: KDD (2014)
20. Yuan, S., Wang, J., Zhao, X.: Real-time bidding for online advertising: measurement and analysis. In: ADKDD (2013)
21. Zhang, W., Wang, J.: Statistical arbitrage mining for display advertising. In: KDD (2015)
22. Zhang, W., Yuan, S., Wang, J.: Optimal real-time bidding for display advertising. In: KDD (2014)