Scalable Bid Landscape Forecasting in Real-time Bidding

Aritra Ghosh*[1] (🖂), Saayan Mitra[2], Somdeb Sarkhel[2], Jason Xie[3], Gang Wu[2], and Viswanathan Swaminathan[2]

[1] University of Massachusetts, Amherst, MA, USA
[2] Adobe Research, San Jose, CA, USA
[3] Adobe Advertising Cloud, Emeryville, CA, USA
arighosh@cs.umass.edu, {smitra,sarkhel,jasonxie, gawu,vishy}@adobe.com

Abstract. In programmatic advertising, ad slots are usually sold using second-price (SP) auctions in real-time. The highest bidding advertiser wins but pays only the second highest bid (known as the *winning price*). In SP, for a single item, the dominant strategy of each bidder is to bid the true value from the bidder's perspective. However, in a practical setting, with budget constraints, bidding the true value is a sub-optimal strategy. Hence, to devise an optimal bidding strategy, it is of utmost importance to learn the winning price distribution accurately. Moreover, a demand-side platform (DSP), which bids on behalf of advertisers, observes the winning price if it wins the auction. For losing auctions, DSPs can only treat its bidding price as the lower bound for the unknown winning price. In literature, typically censored regression is used to model such partially observed data. A common assumption in censored regression is that the winning price is drawn from a fixed variance (homoscedastic) uni-modal distribution (most often Gaussian). However, in reality, these assumptions are often violated. We relax these assumptions and propose a heteroscedastic fully parametric censored regression approach, as well as a mixture density censored network. Our approach not only generalizes censored regression but also provides flexibility to model arbitrarily distributed real-world data. Experimental evaluation on the publicly available dataset for winning price estimation demonstrates the effectiveness of our method. Furthermore, we evaluate our algorithm on one of the largest demand-side platform and significant improvement has been achieved in comparison with the baseline solutions.

Keywords: Computational Advertising · Real-time Bidding · Censored Regression · Bid Landscape Forecasting

1 Introduction

Real-time Bidding (RTB) has become the dominant mechanism to sell ad slots over the internet in recent times. In RTB, ad display opportunities are auctioned

* This work was conducted while the first author was doing an internship at Adobe Research, USA

when available from the publishers (sellers) to the advertisers (buyers). When a user sees the ad that won the auction, it is counted as an *ad impression*. An RTB ecosystem consists of supply-side platforms (SSP), demand-side platforms (DSP) and an Ad Exchange. When a user visits a publisher's page, the SSP sends a request to the Ad Exchange for an ad display opportunity which is then rerouted to DSPs in the form of a bid request. DSPs bid on behalf of the advertisers at the Ad Exchange. The winner of the auction places the Ad on the publisher's site. Ad Exchanges usually employ second-price auction (SP) where the winning DSP only has to pay the second highest bidding price [21]. Since this price is the minimum bidding price DSP needs to win, it is known as the *winning price*. When a DSP wins the auction, it knows the actual winning price. However, if the DSP loses the auction, the Ad Exchange does not reveal the winning price. In that case, the bidding price provides a lower bound on the winning price. This mixture of observed and partially-observed (lower bound) data is known as *right censored data*. The data to the *right* of the bidding price is not observed since it is right censored.

For a single *ad impression* under the second price auction scheme, the dominant strategy for an advertiser is to bid the true value of the ad. In this scenario, knowing the bidding prices of other DSPs does not change a bidder's strategy [6]. However, in reality, DSPs have budget constraints with a utility goal (e.g., number of impressions, clicks, conversions). Under budget constraints, with repeated auctions, bidding the true value is no longer the dominant strategy [3]. In this setting, knowledge of the bidding prices of other bidders can allow one to change the bid to improve its expected utility. DSP needs to estimate the cost and utility of an auction to compute the optimal bidding strategy (or bidding price) [23]. To compute the expected cost as well as the expected utility one needs to know the winning price distribution. Therefore, modeling the winning price distribution is an important problem for a DSP [12]. This problem is also referred to as the *Bid landscape forecasting* problem.

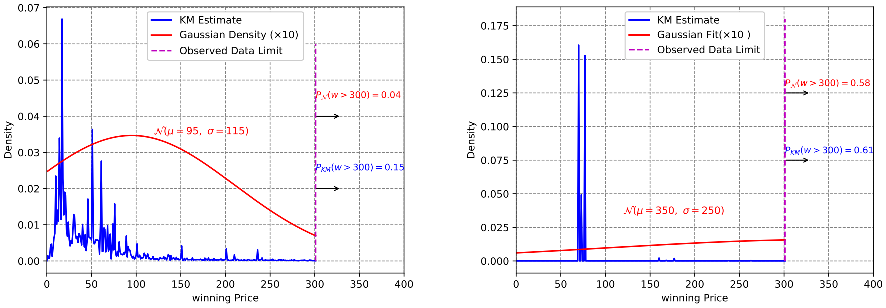
Learning the bid landscape from a mix of observed and partially-observed data poses a real challenge. It is not possible for DSPs to know the behavior of the winning price beyond the maximum bidding price. Parametric approaches often assume that the winning price follows some distribution. In the existing literature, Gaussian and Log-Normal distributions are often used for modeling the winning price [20, 5]. However, these simple distributions do not always capture all the complexities of real-world data. Moreover, for losing bids, the density of winning price cannot be measured directly, and hence a standard log-likelihood based estimate does not typically work on the censored data. In this scenario, a common parametric method used is *Censored Regression*, which combines the log density and the log probability for winning and losing auctions respectively [20, 13]. Another common alternative is to use non-parametric survival based methods using the Kaplan-Meier (KM) estimate for censored data [10]. To improve the performance of the KM estimate, clustering the input is important. Interestingly, in [18], the authors proposed to grow a decision tree based on survival methods. In the absence of distributional assumptions, non-parametric methods (KM) work well. However, efficiently scaling non-parametric methods is also challenging. On the other hand, parametric methods work on strong distributional assumptions. When the assumptions are violated, inconsistency arises. For a general discussion of the censored problem in machine learning, readers are referred to [16].

Learning a distribution is generally more challenging than point estimation. Thus, parametric approaches in previous research often considered point estimation [20, 19]. However, to obtain an optimal bidding strategy, one needs the distribution of the winning price. On the other hand, non-parametric approaches like the KM method computes the distribution without any assumptions. However, these methods require clustering the data to improve the accuracy of the model using some heuristics. Clustering based on feature attributes makes these methods sub-optimal impacting generalization ability for dynamic real-world ad data.

In this paper, we close the gap of violated assumptions in parametric approaches on censored data. Censored regression-based approaches assume a unimodal (often Gaussian) distribution on winning price. Additionally, it assumes that the standard deviation of the Gaussian distribution is unknown but fixed. However, in most real-world datasets these assumptions are often violated. For example, in Figure 1, we present two winning price distributions (learned using the KM estimate) as well as fitted Gaussian distributions[4] on two different partitions of the iPinYou dataset [24]. It is evident from Figure 1 that the distributions are neither Gaussian (blue line) nor have fixed variance (red line). In this paper, we relax each of these assumptions one by one and propose a general framework to solve the problem of predicting the winning price distribution using partially observed censored data. We first propose an additional parameterization which addresses the fixed variance assumption. Further, the Mixture Density Network is known to approximate any continuous, differentiable function with enough hidden nodes [4]. We propose a Mixture Density Censored Network to learn smooth winning price distribution using the censored data. We refer to it as MCNet in the rest of the paper. Both of our proposed approaches are generalizations of the Censored Regression.

Our main contributions are as follows. The typical deployed system uses Censored regression for point estimation of the winning price. However, we argue that point estimation is not enough for an optimal bidding strategy. We improve upon the parametric Censored Regression model to a general framework under minimal assumptions. We pose Censored Regression as a solution to the winning price distribution estimation problem (instead of a point estimate). To the best of our knowledge, we are the first to apply the mixture density network on censored data for learning the arbitrary distribution of the winning price. Our

[4] We fit the unimodal Gaussian minimizing KL divergence with the estimated KM distribution. We would like to point out, although in Figure 1(b) winning price density is unimodal (within the limit), the probability of winning price beyond the max bid price is high (0.61). Thus the fitted Gaussian has a mean of 350 and std dev of 250 further from the peak at 75.



(a) Cluster 1

(b) Cluster 2

Fig. 1: KM Estimate and Gaussian Fit on two clusters of Session-2 date 2013-06-12 on iPinYou [24]

extensive experiments on a real-world public dataset show that our approach vastly out-performs existing state-of-the-art approaches such as [20]. Evaluation on the historical bid data from Adobe (DSP) shows the efficacy of our scalable solution. While we restricted the analysis to winning price distribution in real-time bidding, MCNet is applicable to any partially observed censored data problem.

2 Background & Related Work

In RTB, a DSP gets bid requests from the Ad exchange. We represent the i^{th} bid request by a feature vector $\mathbf{x}_i$, which captures all the characteristics of the bid request. Most of the elements of $\mathbf{x}_i$'s are categorical (publisher verticals, user's device, etc.). If DSP wins the auction, it pays the second (winning) price. Formally, the winning price is,

$$\mathbf{w}_i = \max\{\mathbf{b}_i^{\text{Pub}}, \mathbf{b}_i^{\text{DSP}_1}, \mathbf{b}_i^{\text{DSP}_2}, \cdots, \mathbf{b}_i^{\text{DSP}_K}\}$$

where $\mathbf{b}_i^{\text{Pub}}$ is the floor price set by the publisher[5] (often 0), and $\mathbf{b}_i^{\text{DSP}_1}, \cdots, \mathbf{b}_i^{\text{DSP}_K}$ are bidding prices from all other participating DSPs. We use $\mathbf{b}_i$ to denote the bidding price from the DSP of our interest. Here we provide an example to illustrate the winning price (in SP auction). Suppose DSPs A, B, C bid \$1, \$2, \$3 respectively for a bid request. DSP C then wins the auction and pays the second-highest price, i.e., \$2. For DSP C, the winning price is \$2 (observed). For losing DSPs, A, and B, the winning price is \$3 (which is unknown to them). In this paper, we define the winning price from the perspective of a single DSP.

[5] For simplicity, we view the floor price by the publisher as a bid from an additional DSP.

Learning the landscape of winning price accurately is important for an optimal bidding strategy. A DSP is usually interested in some utility $\mathbf{u}_i$ (e.g., clicks, impressions, conversions) for each bid request $\mathbf{x}_i$ and wants to maximize the overall utility using bidding strategy $\mathcal{A}$ and with budget $\mathcal{B}$. This can be represented by the following optimization problem, $\max_{\mathcal{A}} \sum_i \mathbf{u}_i$ s.t. $\sum_i cost_i \leq \mathcal{B}$, where $cost_i$ is the price the DSP pays, if it wins the auction. Although the variables are unknown beforehand, the expected cost and the utility can be computed using the historical bid information. Thus the problem simplifies to,

$$\max_{\mathcal{A}} \sum_i \mathrm{E}[\mathbf{u}_i|\mathbf{x}_i, \mathbf{b}_i] \text{ s.t. } \sum_i \mathrm{E}[cost_i|\mathbf{x}_i, \mathbf{b}_i] \leq \mathcal{B} \tag{1}$$

Note that, the expected utility $\mathbf{u}_i$ is conditioned on bid request $\mathbf{x}_i$ and the actual bid $\mathbf{b}_i$. For bid request $\mathbf{x}_i$, we represent the winning price distribution as $P_{\mathbf{w}}(\mathbf{W}_i|\mathbf{x}_i)$, and its cumulative distribution function (cdf) as $F_{\mathbf{w}}(\mathbf{W}_i|\mathbf{x}_i)$. If the DSP bids $\mathbf{b}_i$ for $\mathbf{x}_i$, expected cost and expected utility (for SP auction) is,

$$\mathrm{E}[cost_i|\mathbf{x}_i, \mathbf{b}_i] = \int_0^{\mathbf{b}_i} \mathbf{w} P_{\mathbf{w}}(\mathbf{W}_i = \mathbf{w}|\mathbf{x}_i) d\mathbf{w}, \quad \mathrm{E}[\mathbf{u}_i|\mathbf{x}_i, \mathbf{b}_i] = F_{\mathbf{w}}(\mathbf{b}_i|\mathbf{x}_i)\mathrm{E}[\mathbf{u}_i|\mathbf{x}_i]$$

An example of expected utility conditioned on bid request ($\mathrm{E}[\mathbf{u}_i|\mathbf{x}_i]$) is Click-through rate (CTR). CTR prediction is a well-studied problem in academia and the industry [17]. We want to point out that the expected cost ($\mathrm{E}[cost_i|\mathbf{x}_i, \mathbf{b}_i]$) is not the same as the expected winning price ($\mathrm{E}[\mathbf{W}_i|\mathbf{x}_i]$). The former is always lower than the latter and is equal only when $\mathbf{b}_i \to \infty$ (i.e., when the advertiser wins the auction with probability 1 and observe the winning price). Thus predicting the winning price distribution instead of the point estimate is important [15]. Further, for pacing the budget, one requires an estimate of winning price distribution [2]. In [25], the authors proposed an unbiased learning algorithm of click-through rate estimation using the winning price distribution. Earlier parametric methods, considered point estimation of the winning price. The censored regression-based approach assumes a standard unimodal distribution with a fixed but unknown variance to model the winning price [20, 26, 17]. In another paradigm, non-parametric methods such as the KM estimator has been successful for modeling censored data [10, 18].

In the rest of the paper, we use P to denote probability density function (pdf) and Pr to denote the usual probabilities. Next, we describe how Censored Regression is applied to model the winning price.

2.1 Censored Regression

The data available to DSP is right censored by the Ad Exchange, i.e., for losing bids only a lower bound (the bidding price) of the winning price is known. However, a maximum likelihood estimator (MLE) can still work on the censored data with some assumptions.

In [20], the authors assume that the winning price follows a normal distribution with fixed but unknown variance σ. The authors assume a linear relationship between the mean of the normal distribution and the input feature vector.

We use $\mathbf{W}_i$ to represent the random variable of winning price distribution of i^{th} bid request whereas $\mathbf{w}_i$ is the realization of that. Thus $\mathbf{w}_i = \beta^T \mathbf{x}_i + \epsilon_i$ where ϵ_i are independent and identically distributed ($i.i.d$) from $\mathcal{N}(0, \sigma^2)$ and $\mathbf{W}_i \sim \mathcal{N}(\beta^T \mathbf{x}_i, \sigma^2)$.

One can use any standard distribution in the censored regression approach. In [19], the authors argue that maximal bidding price in the limit (of infinite DSPs) resembles Gumbel distribution. However, for the generality of learning from censored data, we do not constrain on any particular distribution in this paper. Moreover, the linear link function can be replaced with any non-linear function. Thus, $\mathbf{w}_i$ can be parameterized as $\mathbf{w}_i = f(\beta, \mathbf{x}_i) + \epsilon_i$ where f can be any continuous differentiable function. With the success of deep models, in [19], the authors parameterize $f(\beta, \mathbf{x}_i)$ with a deep network for additional flexibility. Since we know the winning price for winning auctions, likelihood is simply the probability density function (pdf) $P(\mathbf{W}_i = \mathbf{w}_i) = \frac{1}{\sigma}\phi(\frac{\mathbf{w}_i - \beta^T \mathbf{x}_i}{\sigma})$ where ϕ is the pdf of standard normal $\mathcal{N}(0, 1)$. Note that, $\mathbf{W}_i$ is the random variable associated with the winning price distribution whereas $\mathbf{w}_i$ is the observed winning price. For losing auctions, as we do not know the winning price, the pdf is unknown to us. However, from the lower bound on the winning price, we can compute the probability that bid $\mathbf{b}_i$ will lose in the auction for bid request $\mathbf{x}_i$, under the estimated distribution of $\mathbf{W}_i$ as $\Pr(\mathbf{W}_i > \mathbf{b}_i) = \Pr(\epsilon_i < \beta^T \mathbf{x}_i - \mathbf{b}_i) = \Phi(\frac{\beta^T \mathbf{x}_i - \mathbf{b}_i}{\sigma})$. Here Φ is the cdf for standard normal distribution. As discussed, ϕ and Φ can be replaced with pdf and cdf of any other distribution (with different parameterization).

Taking log of the density for winning auctions $\mathcal{W}$ and the log-probability for losing auctions $\mathcal{L}$, we get the following objective function [20],

$$\beta^*, \sigma^* = \arg\max_{\beta, \sigma > 0} \sum_{i \in \mathcal{W}} \log\left(\frac{1}{\sigma}\phi(\frac{\mathbf{w}_i - \beta^T \mathbf{x}_i}{\sigma})\right) + \sum_{i \in \mathcal{L}} \log\left(\Phi(\frac{\beta^T \mathbf{x}_i - \mathbf{b}_i}{\sigma})\right) \quad (2)$$

When the ϵ_i (noise in the winning price model) are i.i.d samples from a fixed variance normal distribution, censored regression is an unbiased and consistent estimator [9, 8].

3 Methodology

In this paper, we build on top of (Gaussian) censored regression-based approach by relaxing some of the assumptions that do not hold in practice. First, we relax the assumption of *homoscedasticity*, i.e., noise (or error) follows a normal distribution with fixed but possibly unknown variance, by modeling it as a fully parametric censored regression. Then we also relax the unimodality assumption by proposing a mixture density censored network. We describe the details of our approaches in the next two subsections.

3.1 Fully Parametric Censored Regression

The censored regression approach assumes that the winning price is normally distributed with a fixed standard deviation. As we discussed, in Figure 1, the variance of the fitted Gaussian model is not fixed. If the noise ϵ is heteroscedastic or not from a fixed variance normal distribution, the MLE is biased and inconsistent. Using a single σ to model all bid requests, will not fully utilize the predictive power of the censored regression model. Moreover, while the point estimate (mean) of the winning price is not dependent on the estimated variance, the *Bid landscape* changes with σ. We remove the restriction of homoscedasticity in the censored regression model and pose it as a solution to the distribution learning problem.

Specifically, we assume the error term ϵ is coming from a parametric distribution conditioned on the features. This solves the problem of error term coming from fixed variance distribution. We assume the noise term ϵ_i is coming from $\mathcal{N}(0, \sigma_i^2)$ where $\sigma_i = \exp(\alpha^T \mathbf{x}_i)$.

The likelihood for winning the auction is, $P(\mathbf{W}_i = \mathbf{w}_i) = \frac{1}{\exp(\alpha^T \mathbf{x}_i)} \phi(\frac{\mathbf{w}_i - \beta^T \mathbf{x}_i}{\exp(\alpha^T \mathbf{x}_i)})$ where the predicted random variable $\mathbf{W}_i \sim \mathcal{N}(\beta^T \mathbf{x}_i, \exp(\alpha^T \mathbf{x}_i)^2)$ and ϕ is the pdf of $\mathcal{N}(0,1)$. In fully parametric censored regression, $\epsilon_i \sim \mathcal{N}(0, \exp(\alpha^T \mathbf{x}_i)^2)$ are not *i.i.d* samples. For losing bids, we can similarly compute the probability based on the lower bound (bidding price $\mathbf{b}_i$)

$$\Pr(\mathbf{W}_i > \mathbf{b}_i) = P(\epsilon_i < \beta^T \mathbf{x}_i - \mathbf{b}_i) = \Phi(\frac{\beta^T \mathbf{x}_i - \mathbf{b}_i}{\exp(\alpha^T \mathbf{x}_i)})$$

Under the assumption of normal but varying variance on the noise, we can still get a consistent and unbiased estimator by solving the following problem.

$$\beta^*, \alpha^* = \arg\max_{\beta,\alpha} \sum_{i \in \mathcal{W}} \log\left(\frac{1}{\exp(\alpha^T \mathbf{x}_i)} \phi(\frac{\mathbf{w}_i - \beta^T \mathbf{x}_i}{\exp(\alpha^T \mathbf{x}_i)})\right) + \sum_{i \in \mathcal{L}} \log\left(\Phi(\frac{\beta^T \mathbf{x}_i - \mathbf{b}_i}{\exp(\alpha^T \mathbf{x}_i)})\right) \tag{3}$$

3.2 Mixture Density Censored Network (MCNet)

In the previous subsection, we relaxed the fixed variance problem by using a parametric σ. However, no standard distribution can model the multi-modality that we observe in real-world data. For example, in Figure 1(b), we see mostly unimodal behavior below the max bid price. However, the probability of losing an auction is often high (61% in Figure 1(b)). Thus even with parametric standard deviation, when we minimize the KL-divergence with a Gaussian, the mean shifts towards the middle. Inspired by the Gaussian Mixture Model (GMM) [4] we propose a Mixture Density Censored Network (MCNet). MCNet resembles a Mixture Density Network while handling partially observed censored data for learning arbitrary continuous distribution.

In a GMM, the estimated random variable $\mathbf{W}_i$ consists of K Gaussian densities and has the following pdf, $P(\mathbf{W}_i = \mathbf{w}_i) = \sum_{k=1}^{K} \frac{\pi_k(\mathbf{x}_i)}{\sigma_k(\mathbf{x}_i)} \phi(\frac{\mathbf{w}_i - \mu_k(\mathbf{x}_i)}{\sigma_k^2(\mathbf{x}_i)})$. Here $\pi_k(\mathbf{x}), \mu_k(\mathbf{x}), \sigma_k(\mathbf{x})$ are the weight, mean and standard deviation for k^{th} mixture density respectively where $k \in \{1, \cdots, K\}$. To model the censored regression problem as a mixture model, a straightforward way is to formulate the mean of the Gaussian distributions with a linear function. Furthermore, to impose positivity of σ, we model the logarithm of the standard deviation as a linear function. We impose a similar positivity constraint on the weight parameters. The parameters of the mixture model are (for $k \in \{1, \cdots, K\}$),

$$\mu_k(\mathbf{x}_i) = \beta_{\mu,k}^T \mathbf{x}_i,\ \sigma_k(\mathbf{x}_i) = \exp(\beta_{\sigma,k}^T \mathbf{x}_i),\ \pi_k(\mathbf{x}_i) = \frac{\exp(\beta_{\pi,k}^T \mathbf{x}_i)}{\sum_{j=1}^{K} \exp(\beta_{\pi,j}^T \mathbf{x}_i)}$$

We can further generalize this mixture model and define a Mixture Density Network (MDN) by parameterizing $\pi_k(\mathbf{x}_i), \mu_k(\mathbf{x}_i), \sigma_k(\mathbf{x}_i)$ with a deep network. In applications such as speech and image processing and astrophysics, MDNs have been found useful [22, 14]. MDN can work with any reasonable choice of base distribution.

MDN combines mixture models with neural networks. The output activation layer, consists of $3K$ nodes ($z_{i,k}$ for $i \in \{\mu, \sigma, \pi\}$ and $k \in \{1, \cdots, K\}$). We use $z_{\mu,k}, z_{\sigma,k}, z_{\pi,k}$ to retrieve the mean, standard deviation and weight parameters of k^{th} density,

$$\mu_k(\mathbf{x}_i) = z_{\mu,k}(\mathbf{x}),\ \sigma_k(\mathbf{x}_i) = \exp(z_{\sigma,k}(\mathbf{x})),\ \pi_k(\mathbf{x}_i) = \frac{\exp(z_{\pi,k}(\mathbf{x}_i))}{\sum_{j=1}^{K} \exp(z_{\pi,j}(\mathbf{x}_i))} \quad (4)$$

MDN outputs conditional probabilities that are used for learning distribution from fully observed data [4]. For the censored problem, however, we only observe partial data. We can now extend MDN to MCNet on censored data. Instead of conditional output probabilities, MCNet outputs the probability of losing in case auction is lost. Thus, we can compute the log-likelihood function on partially observed data. Taking the likelihood for winning auctions, the corresponding negative log-likelihood for all the winning auctions is given by $\sum_{i \in \mathcal{W}} -\log(\sum_{k=1}^{K} \frac{\pi_k(\mathbf{x}_i)}{\sigma_k(\mathbf{x}_i)} \phi(\frac{\mathbf{w}_i - \mu_k(\mathbf{x}_i)}{\sigma_k(\mathbf{x}_i)}))$ where, ϕ is the pdf of $\mathcal{N}(0, 1)$. For losing bids, we can similarly compute the probability of losing based on the lower bound, $\Pr(\mathbf{W}_i > \mathbf{b}_i) = \sum_{k=1}^{K} \pi_k(\mathbf{x}_i) \varPhi(\frac{\mu_k(\mathbf{x}_i) - \mathbf{b}_i}{\sigma_k(\mathbf{x}_i)})$

Negative log-probability of all the losing auctions from the mixture density is,

$$\sum_{i \in \mathcal{L}} -\log(\sum_{k=1}^{K} \pi_k(\mathbf{x}_i) \varPhi(\frac{\mu_k(\mathbf{x}_i) - \mathbf{b}_i}{\sigma_k(\mathbf{x}_i)})) \quad (5)$$

where, $\varPhi$ represents the cdf of $\mathcal{N}(0, 1)$.

From Figure 1, recall that the distribution is not unimodal and has multiple peaks. To address the multi-modality of the data we used a mixture of multiple densities. The embedded deep network in the MCNet ($\mathcal{M}$) is trained to learn the mean and standard deviation parameters of each of the constituents of the mixture model as well as the corresponding weights. Combining all the auctions, we get the following optimization function for censored data,

$$\begin{aligned}\mathcal{M}^* = \arg\max_{\mathcal{M}} \sum_{i\in\mathcal{L}} \log(\sum_{k=1}^{K} \pi_k(\mathbf{x}_i)\Phi(\frac{\mu_k(\mathbf{x}_i) - \mathbf{b}_i}{\sigma_k(\mathbf{x}_i)})) \\ + \sum_{i\in\mathcal{W}} \log(\sum_{k=1}^{K} \frac{\pi_k(\mathbf{x}_i)}{\sigma_k(\mathbf{x}_i)}\phi(\frac{\mathbf{w}_i - \mu_k(\mathbf{x}_i)}{\sigma_k(\mathbf{x}_i)}) \qquad (6)\end{aligned}$$

where $\mathcal{M}$ is the neural network (parameters).

3.3 Optimization

It is easy to compute gradients of Eq. 2, 3, 6 with respect to all the parameters. We used Adam optimizer for stochastic gradient optimization [11].

4 Experimental Results

In this section, we discuss experimental settings, evaluation measures, and results.

4.1 Experimental Settings

Datasets: We ran experiments on the publicly available iPinYou dataset [24] as well as on a proprietary dataset collected from Adobe Adcloud (a DSP). The iPinYou dataset contains censored winning price information. Further experimentation was done on a sampled week's data from Adobe Adcloud. iPinYou data is grouped into two subsets: session 2 (dates from 2013-06-06 to 2013-06-12), and session 3 (2013-10-19 to 2013-10-27). We experimented with the individual dates within sessions as well. For all the datasets, we allocated 60% for training, 20% for validation and rest 20% for testing. We report the average as well as the standard deviation over five iterations. Similar to previous research on the iPinYou dataset, we remove fields that are not directly related to the winning price at the onset [20, 18]. The fields used in our methods are UserAgent, Region, City, AdExchange, Domain, AdSlotId, SlotWidth, SlotHeight, SlotVisibility, SlotFormat, Usertag. Every categorical feature (e.g City), is one-hot encoded, whereas every numerical feature (e.g Ad height) is categorized into bins and subsequently represented as one-hot encoded vectors. This way, each bid request is represented as a high-dimensional sparse vector. Table 1 shows the statistics of sessions in the iPinYou datasets. The number of samples and win rates for individual dates are mentioned in Table. 2.

Table 1: Basic statistics of iPinYou Sessions

Session	sample	feature	win rate (%)
2	53,289,330	40,664	22.87
3	10,566,743	25,020	29.64

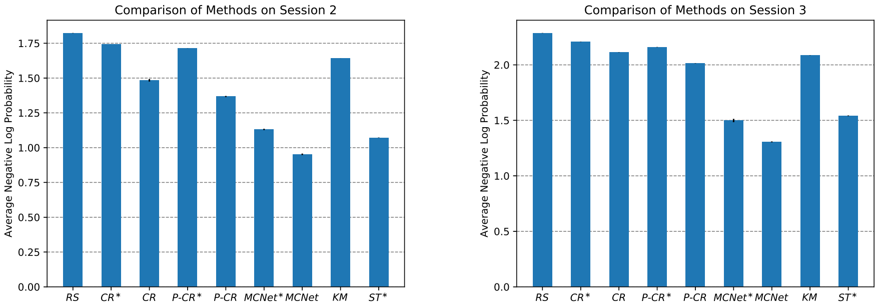
Evaluation Settings: Evaluation on partially observed data is difficult when the winning price is unknown especially for point estimation. In [20], the authors simulated new synthetic data from the original winning auctions. While the added censoring allows validating point estimate, it does not use the whole data (or the true distribution). We evaluate the performance of predicting the winning price distribution rather than the point estimate itself. Thus we use the whole data without generating simulated censoring behavior. This setting is similar to earlier work on the survival tree-based method where the authors evaluated predicting the distribution and used the original data [18].

Parametric methods: We compared the Censored Regression (CR) approach with our methods: Fully parametric Censored Regression (P-CR) and Mixture Density Censored Network (MCNet). For every method, we added an L2 regularization term for each weight vector to prevent over-fitting. For MCNet, we added an additional hyper-parameter on the number of mixtures. We chose a fully connected hidden layer with 64 nodes with RelU activation function as the architecture. Our framework is general and can be extended to multiple layers. The number of mixture components was varied from 2-4 for individual dates and 2-8 for the experiments on the two sessions. We used Adam optimizer with a learning rate of 10^{-3}. Mini-batch training was employed due to the high volume of the data and we fixed the batch size to 1024 samples. We employed early stopping on the training loss and do not observe the validation loss for early stopping. This way, all the methods are treated similarly. The L2 regularization was varied from 10^{-6} to 10 (in log scale). We implemented the parametric methods in Tensorflow [1]. For the initialization of weight vectors, we sampled randomly from the standard normal distribution in all our experiments.

Recently extending Censored Regression (CR), in [19], the authors proposed to use deep model (DeepCR) to parameterize the mean to provide more flexibility in the point estimation. Additionally, the authors proposed to use Gumbel distribution for point estimation. Note that, MCNet generalizes the DeepCR model when using only one mixture component and Gumbel as the base distribution. We did not see much improvement when using Gumbel to parameterize mixture components with our initial experiments. With enough Gaussian mixture components, MCNet can approximate any smooth distribution. As neural architecture is not the primary motivation for this paper, we do not discuss different architectures or distributions in this paper.

Non-parametric methods: To the best of our knowledge, parametric methods and non-parametric methods were not compared together for winning price distribution estimation in earlier research. We compared our approaches with non-parametric approaches based on Kaplan-Meier (KM) estimate and the Survival tree (ST) method. The KM and ST based methods produce winning price distributions until the maximum bid price since the winning price behavior above that is unknown. To represent a complete landscape with the probability distribution summing to one, we introduce an extra variable representing the event that the winning price is beyond the maximum bid price. For the Survival tree, we varied the tree height from 1-20.

In the ST method, the Survival tree is built by running an Expectation Maximization (EM) algorithm for each field to cluster similar attributes. If data has F fields and the average number of attributes in each field is K, then for n samples, the EM algorithm takes, $\mathcal{O}(FKln)$ steps to cluster features based on their density for l iterations. With depth d, total complexity becomes $\mathcal{O}(FKlnd)$. Given this runtime, we could not run ST using all attributes of *Domain*, *SlotId* fields (these fields were removed in previous research [18]). We trimmed the *Domain* and *SlotId* features by combining the attributes that appeared less than 10^3 times. We created *"other domains"* and *"other slot ids"* bins for these less frequent attributes. This improved the time complexity and made the method viable. But for the CR-based methods, we could easily relax this threshold and trimmed both the features where the attributes appeared less than 10 times in the dataset in either session. For a fair comparison, whenever we use the same feature trimming in the parametric methods as ST, we denote using $\text{CR}^*, \text{P-CR}^*, \text{MCNet}^*$. Note that parametric methods can scale easily whereas the non-parametric ST method cannot.



(a) Session 2 Performance

(b) Session 3 Performance

Fig. 2: iPinYou session's *ANLP* . Error bar represents the standard deviation.

Baseline Method: We also propose a simple baseline method and compare it with other methods. The baseline algorithm picks a winning price randomly conditioned on the win rate. We denote this as the Random Strategy (RS). Formally, let the maximum bid price be z and probability of a win be p. Then, the probability that the winning price is w is given by

$$P(\mathbf{W} = w) = \frac{p}{z} \text{ if } w \in [0, z], \text{ and } 0 \text{ if } w < 0 \text{ and } \quad \int_{z}^{\infty} \Pr(\mathbf{W} = w) dw = 1 - p$$

Thus with probability $1 - p$, it predicts the event that winning price is greater than max bid price and with probability p it draws from $\mathcal{U}(0, z)$ where $\mathcal{U}$ is the Uniform distribution.

4.2 Evaluation Measure

Our objective is to learn the distribution of the winning price, rather than the point estimate. Hence, we choose Average Negative Log Probability (*ANLP*) as our evaluation measure similar to [18]. *ANLP* is defined as,

$$ANLP \ = -\frac{1}{N}\Big(\sum_{i \in \mathcal{W}} \log \Pr(\mathbf{W}_i = \mathbf{w}_i) + \sum_{i \in \mathcal{L}} \log \Pr(\mathbf{W}_i \geq \mathbf{b}_i)\Big)$$

where $\mathcal{W}$ represents the set of winning auctions, $\mathbf{w}_i$ represents winning price of the i^{th} winning auction, $\mathcal{L}$ is the set of losing auctions, $\mathbf{b}_i$ is the bidding price for the i^{th} losing auction, and $|\mathcal{W}| + |\mathcal{L}| = N$.

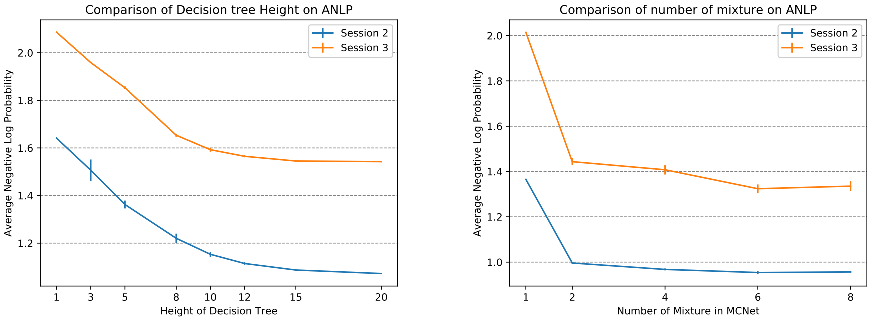
Note that, we computed pdf for winning auctions and probability (or the CDF) for losing auctions while optimizing. While the CDF represents the probability of the event, density does not represent probability. Additionally, bid prices are an integer. The KM method estimates the probability on those discrete points. However, parametric approaches estimate a continuous random variable whose probability at any discrete point is 0. To treat losing bids and winning bids similarly in evaluation, we use quantization trick on the continuous random variable [7]. For the parametric approaches, the estimate $\mathbf{W}_i$ is a continuous random variable. We discretized the random variable $\mathbf{W}_i$ as follows, $\mathbf{W}_i^{\text{bin}} = l$, if $\mathbf{W}_i \in (l - 0.5, l + 0.5]$ where l is an integer. Thus, for winning auctions $\mathcal{W}$ with winning price $\mathbf{w}_i$, quantized probability is,

$$\Pr(\mathbf{W}_i^{\text{bin}} = \mathbf{w}_i) = \Pr(\mathbf{W}_i \leq \mathbf{w}_i + 0.5) - \Pr(\mathbf{W}_i \leq \mathbf{w}_i - 0.5)$$

For losing auctions $\mathcal{L}$, the quantized probability is, $\Pr(\mathbf{W}_i^{\text{bin}} \geq \mathbf{b}_i) = \Pr(\mathbf{W}_i \geq \mathbf{b}_i - 0.5)$. Using quantization technique, winning bids and losing bids are treated similarly for all methods.

4.3 Experimental Results

In this section, we discuss quantitative results on iPinYou sessions 2 and 3. In Table 2, we provide average *ANLP* over different dates as well as the standard deviation (std) numbers. In figure 2, we mention the result on each session as a whole.



(a) Varying Height of ST

(b) Varying mixture components

Fig. 3: Hyper-parameter effect on *ANLP*

Moreover, we plot how number of mixture components as well as tree depth affect the result for MCNet and ST respectively in Figure 3. In sessions 2 and 3 where we include all the dates, we also added the ST method for comparison. As ST did not run with large feature space, we also added CR*, P-CR*, MCNet* for parity (where number of feature was small for all methods).

From Table 2, it is evident, P-CR improves upon CR on most dates (except with low volume dates) asserting the violation of fixed standard deviation assumption. While for P-CR, improvement is around 5%-10%, MCNet improves CR by more than 30% on all dates. Improvement of MCNet re-verify our assumption about the multi-modal nature of the winning price distribution. CR performs better than both RS as well as KM. This is expected as the non-parametric KM estimate does not use any features. However, KM improves RS by around 10% on all dates. ST improves CR and P-CR significantly implying the significance of non-parametric estimators.

In Figure 2, one can see similar trends over CR, P-CR, and MCNet. With feature trimming, MCNet* performs similarly to ST methods. This is expected as both ST and MCNet can predict arbitrary smooth distributions. Although, when the MCNet approach is restricted to fewer features (MCNet*) on the average it performs similarly to ST, the benefits of parametric methods come from the fact that parametric approaches are scalable to large feature as well as input space. It may be observed that the performance of MCNet improves ST by more than 10% on both sessions. While we used only one hidden layer for MCNet, any deep network can be used to parameterize the mixture density network for potentially improving the MCNet results even further.

In Figure 3(a), we plot *ANLP* for different depths of the decision trees. It can be observed that for ST, the performance saturates around depth 15. In Figure 3(b), we also show how the varying number of mixture components impacts *ANLP* . On the larger dataset of Session 2, *ANLP* stabilizes to a low value at 4

Table 2: *ANLP* on Session 2 and 3 individual dates. We report std only if it is higher than 0.01

Date	≈n(×10^6)	wr(%)	RS	CR	p-CR	MCNet	KM	ST
2013-06-06	9.58	18.93	1.55	1.212	1.081	**0.756**	1.403	0.956
2013-06-07	11.13	16.16	1.35	1.036	0.913	**0.641** ± 0.02	1.211	0.823
2013-06-08	5.22	31.17	2.37	1.946	1.695	**1.311** ± 0.02	2.131	1.527
2013-06-09	11.88	13.85	1.17	0.887	0.784	**0.574** ± 0.03	1.071	0.710
2013-06-10	5.61	34.06	2.55	2.130	1.809	**1.252** ± 0.06	2.234	1.502
2013-06-11	5.09	34.13	2.56	2.128	1.810	**1.351**	2.248	1.552
2013-06-12	4.75	34.68	2.59	2.189	1.914	**1.364** ± 0.04	2.273	1.572
2013-10-19	0.35	64.58	4.31	4.135	4.285 ± 0.08	**2.791** ± 0.05	3.659	3.056 ± 0.02
2013-10-20	0.32	65.48	4.33	4.167	4.287 ± 0.15	**2.768** ± 0.1	3.737	3.159
2013-10-21	1.54	54.59	3.77	3.466	3.515	**2.338** ± 0.03	3.272	2.529
2013-10-22	1.21	56.00	3.85	3.641	3.569	**2.576** ± 0.03	3.428	2.779
2013-10-23	1.57	14.30	1.22	1.060	1.033	**0.854** ± 0.02	1.157	0.963
2013-10-24	2.18	11.23	0.985	0.831	0.824	**0.618**	0.904	0.698 ± 0.01
2013-10-25	2.23	14.23	1.21	1.015	0.998	**0.771** ± 0.03	1.131	0.888
2013-10-26	0.53	49.90	3.51	3.432	3.433 ± 0.01	**2.577** ± 0.09	3.228	2.931 ± 0.01
2013-10-27	0.59	18.45	1.53	1.367	1.361	**0.937** ± 0.03	1.348	1.104 ± 0.02

Table 3: *ANLP* on Adobe AdCloud Dataset

	CR	P-CR	MCNet	KM	ST
ANLP	0.4744 ± 0.01	0.4722 ± 0.01	**0.3477** ± 0.01	0.4671 ± 0.01	0.4213 ± 0.02

mixture components. However, for session 3, *ANLP* starts increasing beyond 6 mixture components, implying over-fitting.

Results on Adobe AdCloud Dataset: We also tested our methods on Adobe Advertising Cloud (DSP) offline dataset. We collected a fraction of logs for one week. The number of samples was 31, 772, 122 and the number of features was 33, 492. It had similar categorical as well as real-valued features. We use the same featurization framework and represented each bid request with a sparse vector. In Table 3, we report the *ANLP* results, using the same experimental setup. Note that, MCNet improves CR by 25% while it improves ST by more than 10%. In this dataset, we do see only marginal improvement over using P-CR.

5 Discussion & Future Work

In this paper, we particularly focus on one of the central problems in RTB, the winning price distribution estimation. In practice, DSP depends on the estimated bid landscape to optimize it's bidding strategy. From a revenue perspective, an accurate bid landscape is of utmost importance. While, non-parametric methods can estimate arbitrary distributions, in practice, it is challenging to scale on large datasets. On the other hand, widely used parametric methods, such as Censored Regression in its original form is highly restrictive. We proposed a novel method based on Mixture Density Networks to form a generic framework for estimating arbitrary distribution under censored data. MCNet generalizes a fully parametric Censored regression approach with the number of mixture components set to one. Additionally, Censored regression is a special case of fully parametric censored regression where the standard deviation is fixed. We provided extensive empirical evidence on public datasets and data from a leading DSP to prove the efficacy of our methods. While the mixture of (enough) Gaussian densities can approximate any smooth distribution, further study is needed on the choice of base distribution. A more subtle point arises when learning with censored data as we do not observe any winning price beyond the maximum bidding price. Without any assumptions on the distribution, it is not provably possible to predict the behavior in the censored region. Non-parametric methods only learn density within the limit of maximum bidding price while under strong assumptions of standard distributions, censored regression predicts the behavior of winning price in the censored region. Although MCNet can approximate any smooth distribution, beyond the maximum bidding price, it leads to non-identifiability similar to the KM estimate. It would be interesting to explore combining MCNet with distributional assumptions where the winning price is never observed.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: a system for large-scale machine learning. In: OSDI. vol. 16, pp. 265–283 (2016)
2. Agarwal, D., Ghosh, S., Wei, K., You, S.: Budget pacing for targeted online advertisements at linkedin. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1613–1619. ACM (2014)
3. Balseiro, S.R., Besbes, O., Weintraub, G.Y.: Repeated auctions with budgets in ad exchanges: Approximations and design. Management Science **61**(4), 864–884 (2015)
4. Bishop, C.M.: Mixture density networks. Tech. rep., Citeseer (1994)
5. Cui, Y., Zhang, R., Li, W., Mao, J.: Bid landscape forecasting in online ad exchange marketplace. In: KDD. pp. 265–273. ACM (2011)
6. Edelman, B., Ostrovsky, M., Schwarz, M.: Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. American economic review **97**(1), 242–259 (2007)
7. Gersho, A., Gray, R.M.: Vector quantization and signal compression, vol. 159. Springer Science & Business Media (2012)
8. Greene, W.H.: On the asymptotic bias of the ordinary least squares estimator of the tobit model. Econometrica: Journal of the Econometric Society pp. 505–513 (1981)
9. James, I.R., Smith, P.: Consistency results for linear regression with censored data. The Annals of Statistics pp. 590–600 (1984)
10. Kaplan, E.L., Meier, P.: Nonparametric estimation from incomplete observations. Journal of the American statistical association **53**(282), 457–481 (1958)

11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Lang, K.J., Moseley, B., Vassilvitskii, S.: Handling forecast errors while bidding for display advertising. In: Proceedings of the 21st international conference on World Wide Web. pp. 371–380. ACM (2012)
13. Powell, J.L.: Least absolute deviations estimation for the censored regression model. Journal of Econometrics **25**(3), 303–325 (1984)
14. Salimans, T., Karpathy, A., Chen, X., Kingma, D.P.: Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. arXiv preprint arXiv:1701.05517 (2017)
15. Wang, J., Zhang, W., Yuan, S.: Display advertising with real-time bidding (rtb) and behavioural targeting. arXiv preprint arXiv:1610.03013 (2016)
16. Wang, P., Li, Y., Reddy, C.K.: Machine learning for survival analysis: A survey. arXiv preprint arXiv:1708.04649 (2017)
17. Wang, R., Fu, B., Fu, G., Wang, M.: Deep & cross network for ad click predictions. In: Proceedings of the ADKDD'17. p. 12. ACM (2017)
18. Wang, Y., Ren, K., Zhang, W., Wang, J., Yu, Y.: Functional bid landscape forecasting for display advertising. In: ECML-PKDD. pp. 115–131. Springer (2016)
19. Wu, W., Yeh, M.Y., Chen, M.S.: Deep censored learning of the winning price in the real time bidding. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2526–2535. ACM (2018)
20. Wu, W.C.H., Yeh, M.Y., Chen, M.S.: Predicting winning price in real time bidding with censored data. In: KDD. pp. 1305–1314. ACM (2015)
21. Yuan, S., Wang, J., Zhao, X.: Real-time bidding for online advertising: measurement and analysis. In: Proceedings of the Seventh International Workshop on Data Mining for Online Advertising. p. 3. ACM (2013)
22. Zen, H., Senior, A.: Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. pp. 3844–3848. IEEE (2014)
23. 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. pp. 1077–1086. ACM (2014)
24. Zhang, W., Yuan, S., Wang, J., Shen, X.: Real-time bidding benchmarking with ipinyou dataset. arXiv preprint arXiv:1407.7073 (2014)
25. Zhang, W., Zhou, T., Wang, J., Xu, J.: Bid-aware gradient descent for unbiased learning with censored data in display advertising. In: KDD. pp. 665–674. ACM (2016)
26. Zhu, W.Y., Shih, W.Y., Lee, Y.H., Peng, W.C., Huang, J.L.: A gamma-based regression for winning price estimation in real-time bidding advertising. In: Big Data (Big Data), 2017 IEEE International Conference on. pp. 1610–1619. IEEE (2017)