Real-Time Optimization Of Web Publisher RTB Revenues

Pedro Chahuara
XRCE
pedro.chahuara@xrce.xerox.com

Nicolas Grislain
AlephD
ng@alephd.com

Gregoire Jauvion
AlephD
gj@alephd.com

Jean-Michel Renders
XRCE
jean-michel.renders@xrce.xerox.com

ABSTRACT

This paper describes an engine to optimize web publisher revenues from second-price auctions. These auctions are widely used to sell online ad spaces in a mechanism called real-time bidding (RTB). Optimization within these auctions is crucial for web publishers, because setting appropriate reserve prices can significantly increase revenue. We consider a practical real-world setting where the only available information before an auction occurs consists of a user identifier and an ad placement identifier. The real-world challenges we had to tackle consist mainly of tracking the dependencies on both the user and placement in an highly non-stationary environment and of dealing with censored bid observations. These challenges led us to make the following design choices: (i) we adopted a relatively simple non-parametric regression model of auction revenue based on an incremental time-weighted matrix factorization which implicitly builds adaptive users' and placements' profiles; (ii) we jointly used a non-parametric model to estimate the first and second bids' distribution when they are censored, based on an on-line extension of the Aalen's Additive model.

Our engine is a component of a deployed system handling hundreds of web publishers across the world, serving billions of ads a day to hundreds of millions of visitors. The engine is able to predict, for each auction, an optimal reserve price in approximately one millisecond and yields a significant revenue increase for the web publishers.

CCS CONCEPTS

• **Mathematics of computing → Probability and statistics**; • **Applied computing → Online auctions**;

KEYWORDS

Online learning; Big-Data; Ad-tech; Real-time; Auctions

1 INTRODUCTION

RTB is a mechanism widely used by web publishers to sell their advertisement space. Advertisers bid in an online real-time auction where the one who bids the most wins the space. The auction mechanism typically used is a second-price auction mechanism, where the winning advertiser pays the second bid.

Publishers set a reserve price, also known as the floor price, below which it will not sell the space. This is applied even if all bids received are lower than the reserve price. In this case the space remains unsold. In a winning bid the first bid wins the auction and the publisher revenue is the maximum between the second bid and the reserve price.

If f, b_1 and b_2 are respectively the reserve price, the first and second bids of the auction, the revenue for the publisher is:

$$R(f, b_1, b_2) = \mathbb{1}_{f \leq b_1} max(f, b_2) \tag{1}$$

Most publishers set the reserve price levels manually on a weekly or daily basis. In this paper, we propose an engine that can predict an optimal reserve price in real time for each auction, using the latest most relevant data.

We consider the real-world setting where the only information available when performing the reserve price prediction is the identifier of the internet user and the identifier of the ad placement[1]. Once the auction is over, we observe possibly censored outcomes : if the auction has been won (i.e the reserve price is smaller than the first bid), we observe the first bid and the closing price of the auction, otherwise we do not observe any bid value. There is no censorship if the reserve price is lower than the second bid.

This setting gives two different cases of censorship: (i) if the auction is lost (i.e. no bid above the reserve price), nothing is observed; (ii) if the reserve price is paid (closing price = reserve), then only the first bid is observed. This incomplete information setting occurs frequently in the industry.

Note that we have some kind of "closed-loop" setting when trying to simultaneously learn the optimal reserve price strategy and compensate for the censored information that could result from this strategy. There is an underlying exploration/exploitation trade-off to be controlled, as too high a reserve price could prevent the model from being updated and improved (because of the lack of observed output in the case of censorship), even if it leads to higher revenue, as estimated by the current model.

Finally, a significant advantage of the proposed engine is that its use is not limited to the second-price auction mechanism. It can be extended to any auction mechanism where the outcome of an auction is a straightforward function of the bids. This feature of the engine is useful in practice, as some advertising platforms use non-standard auction mechanisms.

[1]The user and the ad placement correspond respectively to the internet user visiting the web page and to the ad location on the web page.

KDD '17, August 13-17, 2017, Halifax, NS, Canada

ACM ISBN 978-1-4503-4887-4/17/08…$15.00
https://doi.org/10.1145/3097983.3098150

2 RELATED WORK

Maximizing the revenue of online ad spaces has received increasing attention in the machine learning community over the last decade due to the fast growth of the online advertising industry [10].

Most of the work that addresses this problem focuses on generalized second-price auctions with a reserve price since it is the most prevalent mechanism in the industry, used by the major search engines in sponsored search auctions. In this mechanism, bidders compete in an auction for several items.

[?] proposed a method to determine optimal reserve prices in sponsored search auctions in a static way, i.e. the reserve prices are not optimized auction per auction but on a large set of auctions. The authors apply reserve prices that are determined per keyword on the Yahoo! platform and show that the optimization results in a significant increase in revenue.

Other works develop dynamic reserve price optimization models in a setting where information about the user and the placement is available, e.g. the device used by the user, her geographical location or the content type of the website. A regression model with this kind of information used as features, is built to predict either the optimal floor or the highest bid (which can be used indirectly to fix an optimal floor). [3] proposed an approach based on gradient boosting decision trees and mixture models in order to estimate the cumulative distribution function of the highest bid according to a set of targeting attributes. [8] proposed a model to estimate the winning bid by means of a Tobit model in order to perform regression with censored data. This is one of the few pieces of works that addresses the problem of censored data which is extremely important in RTB since lost auctions, even when not disclosing the values of the bids, provide valuable information about the interval during which the bids happened. The main goal of these two works is predicting the first bid value, and the authors do not propose a strategy to set the reserve price. From a publisher perspective, however, estimating the optimal reserve price to maximize the revenue is of most interest.

Another family of works tries to directly address the optimization problem by transforming the original loss function (which has very bad properties, being non-convex and non-differentiable) into a surrogate function with better properties. By construction, these methods are parametric. The most prominent piece of work in this family is the one of [6], which tries to maximize the revenue by means of a linear predictor that is found by *DC* (difference of convex functions) programming. In this case, full access to bid values is assumed. Later, [7] described a parametric approach to determine optimal reserve prices in second-price RTB auctions by defining a smoothed revenue function to avoid non-differentiability. Their method directly estimates the optimal floor price through a Expectation-Maximization algorithm that can be used with several regressors. Nevertheless, this approach is not adaptive, does not consider censorship and can hardly be used in an on-line setting.

An important drawback of methods relying on models learned previously on training data, is that they do not perform well in non-stationary environments, which is unfortunately the case in practice. In fact, the conditions in which they operate can become very different from those in which they were trained.

The need of adaptive methods in RTB has led to the application of models based on "Multi-armed Bandits" (MAB) strategies [2, 4]. The model proposed by [4] leverages a contextual MAB model which seems very appropriate when using features, however they do not deal with the uncertainty resulting from censored data.

A simpler, adaptive approach to obtain an optimal reserve price has been proposed by [9]. Here the highest bid is modelled considering a log-normal distribution and, after observing an auction, the model is updated in a Bayesian fashion. However, the approach remains parametric and relies on the log-normal distribution assumption; moreover, it doesn't consider the censorship issue. In the same paper, the authors describe another adaptive method, that maintains an optimal reserve price by increasing it by a small amount when it is lower than the first bid and by decreasing it by another small amount when it is higher than the first bid; this alternative method, despite its simplicity, turned out to give better performance than the one based on the log-normal assumption.

To summarize, the main differentiators of our engine rely on the following features: (i) it tackles the time-varying environment and the cold-start problem with incremental on-line model adaptation; (ii) it uses as features only the identifiers of the user and of the ad placement and it copes with the sparsity problems; (iii) it is able to solve the inherent censorship issues induced by the reserve price selection strategy.

3 DESCRIPTION OF THE PREDICTION ENGINE

3.1 Problem Statement

The goal of the proposed method is to predict, when there is an auction opportunity for a particular (user, placement) pair but before this auction happens, an optimal reserve price which maximizes the expected revenue. We assume a stream (sequence) of auctions, so that we can incrementally and adaptively learn an optimal strategy from the outcomes of previous auctions.

For a given auction and a particular (user,placement) pair, the optimal *a posteriori* strategy is to set a floor right below the *first bid*:

$$\operatorname*{argmax}_f \mathbb{1}_{f \leq b_1} max(f, b_2) = b_1$$

But the *a priori* optimal floor is far from being the *expected first bid*, because the cost of not perfectly predicting the first bid is very skewed – setting a floor right below the first bid is not costly as opposed to setting a floor right above the first bid which leads to a revenue equal to 0. In other words,

$$\operatorname*{argmax}_f \mathbb{E}\left(\mathbb{1}_{f \leq b_1} max(f, b_2)\right) \neq \mathbb{E}\left(b_1\right)$$

Choosing the expected first bid would be equivalent to minimizing a quadratic cost function which is completely different from the actual cost profile.

For this reason, we have chosen to adopt a non-parametric approach and model, for each (user,placement) pair, the whole revenue profile (in expectation) instead of modelling the sole optimal floor. By "revenue profile", we mean the revenue curve in function of the floor value. By "non-parametric" model, we mean a model which does not rely on a pre-specified class of functions (to model the dependence of the revenue with respect to the reserve price), or to a predetermined distribution or family of distributions (to model the bid distributions). In our case, non-parametric models are built from a discretization of the input space. More precisely, we first discretize the floor space into K bins (or levels) and we model the K values of the revenue for the K reserve prices, $(f^{(1)}, ..., f^{(K)})$ as a function of the user's and placement's profiles. As it will be explained later, the user's and placement's profiles should be understood as "implicit profiles" and will be derived by a latent factor decomposition method.

Latent factor decomposition is commonly used in recommender systems and can be considered as a way to summarize – or to embed – all previous revenue and bid observations related to a particular user or a particular item into a single vector (see [?] for a basic introduction to the method and for its application in recommender systems).

The engine's prediction consists of choosing the floor level that maximizes the expected revenue.

As we will see, updating the whole revenue profile to take into account the outcome of a new auction implies that we can compute the revenue for each floor level, which requires knowing the first and second highest bids (b_1 and b_2). Such information is however not always available: the information is "fully-censored" (neither b_1 nor b_2 is observed) when the selected reserve price is higher than the first bid; the information is "half-censored" (b_1 is observed but not b_2) when the selected reserve price is lower than b_1 but larger than b_2; the information is not censored in the other cases.

The engine tackles censorship by modelling the first and second bid distributions. When the bid information is censored, the engine uses the bid distributions to compute the expected revenue for each floor level; which will then be fed to the revenue profile modeller as a "proxy" of the real one.

A schematic representation of our engine is given in the diagram of Figure 1. In this diagram, u,p and t are used to designate respectively the user, the placement and the current time stamp, while b_1 and b_2 are the first and second bids of the current auction.

We first analyze the deployment constraints that have guided our design choices in 3.2. Next, we describe the revenue profile modeller component in section 3.4, after having introduced some notations and definitions. We then describe the bid distribution modeller component in section 3.5. The section 3.6 briefly describes how to introduce features other than the user and placement identifiers. Finally, section 3.7 assesses the computational complexity of the engine.

3.2 Production constraints

To be deployed in a production environment, i.e. to be able to predict an optimal reserve price auction per auction for a web publisher, the engine must respect several constraints. These constraints significantly influence the prediction methodology.

First, a web publisher typically has several thousand ad placements to sell and possibly several hundreds of millions of different visitors. Given this dimensionality, we need to limit as much as possible the information stored for each placement and internet user. In practice, this sets an upper limit on the size of the users' and placements' profiles.

The average daily number of auctions for a given internet user is typically only a few dozen or less. The engine must therefore be able to perform relevant reserve price predictions after only a few auctions on a user. This sparsity issue gives particular importance to the regularization methodology used.

The bid distributions may vary significantly over time due to the specific behaviour of each advertiser. This behaviour depends, for example, on the data sources used, budget constraints or bidding algorithm. The engine is therefore based on a time-weighted matrix factorization methodology to estimate users' and placement' profiles.

Finally, the reserve price prediction for each auction should be done in about 1 millisecond. The methods described hereafter respect this constraint.

3.3 Observable inputs and notations

The observed features before an auction takes place are the internet user u and the ad placement p identifiers.

We note $\mathcal{D}$ a stream of auctions on which the model will be estimated, and $\mathcal{D}_t$ the set of auctions happening before time t. We denote by $\mathcal{D}_t^u$ and $\mathcal{D}_t^p$ the set of auctions happening before time t involving user u and placement p respectively.

Once the auction happens, we observe the following information depending on the auction outcome: (i) if the auction has been won by an advertiser, we observe the reserve price f, the highest bid b_1 and the publisher revenue; (ii) if no advertiser has won the auction, we only know that the bids are upper-bounded by the reserve price f.

In the rest of this section, an auction will be denoted by a, and the corresponding time, internet user and ad placement will be denoted respectively by t_a, u_a and p_a.

3.4 Building and updating revenue profile models

In a nutshell, the method consists in predicting the revenue for any triplet <user u, placement p, reserve price level $f^{(k)}$> by latent factor decomposition.

More precisely, for $k \in [\![1, K]\!]$, the revenue $R^{(k)}$ when the reserve price $f^{(k)}$ has been set in an auction for an internet user u and an ad placement p is assumed to have the following latent factor decomposition:

$$R^{(k)} = \beta^{(k)} + (X_u^{(k)})' Y_p^{(k)} + \epsilon^{(k)} \tag{2}$$

where $\beta^{(k)}$ is a global bias for reserve price level $f^{(k)}$; $X_u^{(k)}$ and $Y_p^{(k)}$ are latent factors columns (of size L) associated respectively to user u and placement p for reserve price level $f^{(k)}$; $\epsilon^{(k)}$ is the decomposition error term at reserve price level $f^{(k)}$ and is assumed to be gaussian with zero mean and variance σ^2. Note that the prime symbol ($'$) denotes the transpose operator;

Before an auction happens, the expected revenues $R^{(1)}, ..., R^{(K)}$ are predicted, and the reserve price level $f^{(k)}$ which maximizes the expected revenue is chosen for this auction.

3.4.1 Estimation of the latent factors in the off-line case. Suppose that we have observed a set of auctions and their outcome over the

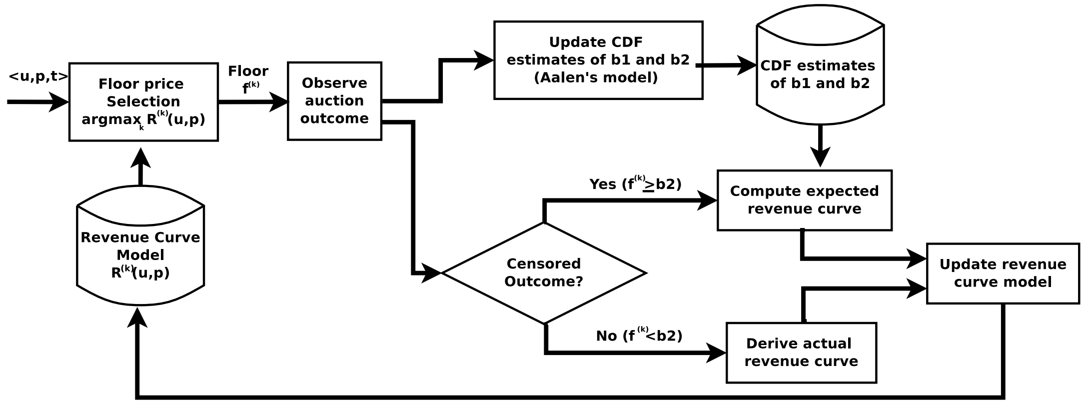

Figure 1: Flow chart of the prediction engine

time interval $[0,T]$, corresponding to the stream $\mathcal{D}_T$. We introduce, for $k \in [\![1,K]\!]$, the following loss function corresponding to the estimation problem over $[0,T]$:

$$L(\hat{\beta}^{(k)}, \hat{X}^{(k)}, \hat{Y}^{(k)}) = \sum_{a\in\mathcal{D}_T} \gamma^{(T-t_a)}(R_a^{(k)} - \hat{\beta}^{(k)} - (\hat{X}_{u_a}^{(k)})'\hat{Y}_{p_a}^{(k)})^2$$
$$+ \sum_{u\in\mathbb{U}} \|\hat{X_u}^{(k)} - X_0^{(k)}\|_\Omega^2 + \sum_{p\in\mathbb{P}} \|\hat{Y_p}^{(k)} - Y_0^{(k)}\|_\Gamma^2 + \|\hat{\beta}^{(k)} - \beta_0^{(k)}\|_\Sigma^2 \tag{3}$$

where:

- the hat symbol ($\hat{.}$) denotes a parameter estimate at time T;
- $\mathbb{U}$ and $\mathbb{P}$ are the sets of all users and placements observed in the data. We note U and P their sizes;
- $\hat{X}^{(k)}$ and $\hat{Y}^{(k)}$ are the matrices of estimated latent factors for users and placements respectively (of size $U \times L$ and $P \times L$);
- γ is a forgetting factor ($\gamma < 1$). A smaller value for the γ hyper-parameter implies a greater time-adaptivity for the latent factors;
- $R_a^{(k)}$ is the "simulated" publisher revenue when setting the k^{th} reserve price level (i.e. $f^{(k)}$) in auction a. If the two highest bids of auction a, b_1 and b_2, are observed, it is simply equal to $\mathbb{1}_{f^{(k)}\leq b_1} max(f^{(k)}, b_2)$, otherwise it is estimated by the method described in section 3.5;
- $\|.\|_A^2$ is the squared Mahalanobis norm with respect to a covariance matrix A
- The last three terms are regularization terms. The hyper-parameters $X_0^{(k)}$ and Ω have a direct bayesian interpretation: assuming that the user latent factors have gaussian priors of mean $X_0^{(k)}$ and covariance matrix Ω', then Ω is nothing else than the covariance matrix scaled by the inverse observation noise variance: $\Omega = \frac{\Omega'}{\sigma^2}$. $Y_0^{(k)}, \Gamma, \beta_0^{(k)}$ and Σ have a similar interpretation.

It is worth noting that the regularization of the problem is more general than the usual formulation where $X_0^{(k)} = Y_0^{(k)} = 0$ and $\Omega = \Gamma = \lambda I$. In particular, biases per user and per placement can be estimated with this formulation. For example, if the i^{th} component of the latent factor $\hat{X_u}^{(k)}$ is dedicated to capturing the bias of user u at level k, the i^{th} component of $Y_0^{(k)}$ will be set to 1 and the i^{th} diagonal value in Γ is set at (approximately) 0. Conversely, if the j^{th} component of the latent factor $\hat{Y_p}^{(k)}$ is dedicated to capturing the bias of placement p at level k, the j^{th} component of $X_0^{(k)}$ will be set to 1 and the j^{th} diagonal value in Ω is set at (approximately) 0.

In the sake of notation simplicity, we drop the (k) superscript here after, knowing that the following equations hold for each level.

The latent factor estimation is performed using an alternating least-squares method, which consists in estimating iteratively the factors associated to users assuming that the factors associated to placements are fixed, and reciprocally. More precisely, it amounts to iterate the following equations: for all $u \in [\![1,U]\!]$, for all $p \in [\![1,P]\!]$, and for all $k \in [\![1,K]\!]$:

$$\begin{aligned}
\hat{X}_u &= X_0 + \left(\sum_{a\in\mathcal{D}_T^u} \gamma^{(T-t_a)}\hat{Y}_{p_a}\hat{Y}'_{p_a} + \Omega^{-1}\right)^{-1} \\
&\quad \left(\sum_{a\in\mathcal{D}_T^u} \gamma^{(T-t_a)}(R_a - \hat{\beta} - X_0'\hat{Y}_{p_a})\hat{Y}_{p_a}\right) \\
\hat{Y}_p &= Y_0 + \left(\sum_{a\in\mathcal{D}_T^p} \gamma^{(T-t_a)}\hat{X}_{u_a}\hat{X}'_{u_a} + \Gamma^{-1}\right)^{-1} \\
&\quad \left(\sum_{a\in\mathcal{D}_T^p} \gamma^{(T-t_a)}(R_a - \hat{\beta} - Y_0'\hat{X}_{u_a})\hat{X}_{u_a}\right) \\
\hat{\beta} &= \beta_0 + \left(\sum_{a\in\mathcal{D}_T} \gamma^{(T-t_a)} + \Sigma^{-1}\right)^{-1} \\
&\quad \left(\sum_{a\in\mathcal{D}_T} \gamma^{(T-t_a)}(R_a - \hat{X}'_{u_a}\hat{Y}_{p_a})\right)
\end{aligned} \tag{4}$$

3.4.2 On-line estimation of the latent factors. To estimate the latent factors on-line, we need to keep in memory the following terms for each user u, placement p and reserve price level k (the superscript k is still omitted):

- $X_{u,cov}$, the time-weighted "covariance" matrix at time T: $X_{u,cov} \doteq \sum_{a\in\mathcal{D}_T^u} \gamma^{(T-t_a)}\hat{Y}_{p_a}\hat{Y}'_{p_a}$. We define symmetrically $Y_{p,cov}$;

– $X_{u,obs}$, which represents the current estimate at time T of $\sum_{a \in \mathcal{D}_T^u} \gamma^{(T-t_a)}(R_a - \hat{\beta} - X_0' \hat{Y}_{p_a})\hat{Y}_{p_a}$. We define symmetrically $Y_{p,obs}$;
– β_{cov} and β_{obs} defined respectively by $\beta_{cov} \doteq \sum_{a \in \mathcal{D}_T} \gamma^{(T-t_a)}$ and $\beta_{obs} \doteq \sum_{a \in \mathcal{D}_T} \gamma^{(T-t_a)}(R_a - \hat{X}_u' \hat{Y}_{p_a})$

With these quantities, it is easy to derive from equation 4 the following update equations, after observing the outcome of auction a: as in alternating least squares, we iterate until convergence and for each reserve price level k:

$$\begin{aligned}
\hat{X}_{u_a} &= X_0 + \left(\gamma^{\Delta t_{u_a}} X_{u_a,cov} + \hat{Y}_{p_a}\hat{Y}_{p_a}' + \Omega^{-1}\right)^{-1} \\
&\quad \left(\gamma^{\Delta t_{u_a}} X_{u_a,obs} + (R_a - \hat{\beta} - X_0'\hat{Y}_{p_a})\hat{Y}_{p_a}\right) \\
\hat{Y}_{p_a} &= Y_0 + \left(\gamma^{\Delta t_{p_a}} Y_{p_a,cov} + \hat{X}_{u_a}\hat{X}_{u_a}' + \Gamma^{-1}\right)^{-1} \\
&\quad \left(\gamma^{\Delta t_{p_a}} Y_{p_a,obs} + (R_a - \hat{\beta} - Y_0'\hat{X}_{u_a})\hat{X}_{u_a}\right) \\
\hat{\beta} &= \beta_0 + \left(\gamma^{\Delta t}\beta_{cov} + 1 + \Sigma^{-1}\right)^{-1} \\
&\quad \left(\gamma^{\Delta t}\beta_{obs} + (R_a - \hat{X}_{u_a}'\hat{Y}_{p_a})\right)
\end{aligned} \tag{5}$$

where Δt, Δt_{u_a} and Δt_{p_a} are respectively the time intervals since the last impression, the last impression for user u_a and the last impression for placement p_a. In practice, one or two iterations are sufficient.

Finally, the following update formulae are applied:

$$\begin{aligned}
X_{u_a,cov} &\leftarrow \gamma^{\Delta t_{u_a}} X_{u_a,cov} + \hat{Y}_{p_a}\hat{Y}_{p_a}' \\
X_{u_a,obs} &\leftarrow \gamma^{\Delta t_{u_a}} X_{u_a,obs} + (R_a^{(k)} - \hat{\beta} - X_0'\hat{Y}_{p_a})\hat{Y}_{p_a} \\
Y_{p_a,cov} &\leftarrow \gamma^{\Delta t_{p_a}} \times Y_{p_a,cov} + \hat{X}_{u_a}\hat{X}_{u_a}' \\
Y_{p_a,obs} &\leftarrow \gamma^{\Delta t_{p_a}} \times Y_{p_a,obs} + (R_a^{(k)} - \hat{\beta} - Y_0'\hat{X}_{u_a})\hat{X}_{u_a} \\
\beta_{cov} &\leftarrow \gamma^{\Delta t}\beta_{cov} + 1 \\
\beta_{obs} &\leftarrow \gamma^{\Delta t}\beta_{obs} + (R_a - \hat{X}_{u_a}'\hat{Y}_{p_a})
\end{aligned} \tag{6}$$

At each iteration and for each reserve price level, there are 2 inversions of $L \times L$ matrices. These inversions do not lead to practical problems because L is chosen relatively low in practice.

Note that using a loss function based on the average least-squares error $\frac{1}{\sum_{a \in \mathcal{D}_T} \gamma^{(T-t_a)}} \sum_{a \in \mathcal{D}_T} \gamma^{(T-t_a)}(R_a^{(k)} - \hat{\beta}^{(k)} - (\hat{X}_{u_a}^{(k)})' \hat{Y}_{p_a}^{(k)})^2$ would lead to much simpler update formulae with no matrix inversion (thanks to the matrix inversion lemma). However, the loss function introduced here is much more adapted to the on-line setting. Indeed, it enables to give a bayesian interpretation of the regularization: a gaussian prior is assigned to each user/placement latent factors. The more observations are available for a user or for a placement, the less the prior impacts the latent factor estimation.

3.5 Dealing with censored observations

When b_1 and b_2 are observed for an auction a, it is easy to compute or, in other words, to simulate the revenue for any floor level: $R_a^{(k)} = \mathbb{1}_{f^{(k)} \le b_1} max(f^{(k)}, b_2)$. Obviously, this could not be done when the bid values are censored. This section describes how we still can estimate $R_a^{(k)}$ with censored observations. Remember that we have two kinds of censorships in the data: half and full censorship (section 3.1)

3.5.1 A brief recap of the Aalen's additive regression model. Let's first recall the Aalen's regression method [1] for left-censored data, that allows us to estimate the cumulative hazard rate of a variable – and, consequently, its cumulative distribution function (CDF) – by means of a set of features (covariates). Note that, for left-censored data, the term "'hazard rate" is an abuse of language, as it is defined here as the ratio of the probability density function, pdf, over the cumulative distribution, cdf, while its standard definition is the ratio $pdf/(1-cdf)$ (the standard case considers right-censored data). Given a discretized variable V with values in the set $(v^{(1)}, ..., v^{(K)})$, n observations of this variable (v_i with $i = 1, \ldots, n$), and a vector C indicating which of these observations are (left) censored (C_i is 1 if observation i is non-censored and 0 when it is censored), the hazard rate of V at a level $v^{(k)}$ can be modeled as a linear combination of $(p+1)$ features represented by the vector $x = (1, x_1, ..., x_p)$ (the first feature aims at capturing the bias which is commonly called the "basis hazard rate" in this framework):

$$\lambda(v^{(k)}|x) = \beta_0^{(k)} + \beta_1^{(k)} x_1 + ... + \beta_p^{(k)} x_p \tag{7}$$

In order to estimate the $((p+1) \times K)$ coefficients $\beta_j^{(k)}$, we basically solve K linear regression problems as follows: for $k = 1, \ldots, K$, we first select the subset S_k of observations with $v_i <= v^{(k)}$ (either censored or not); we then build the corresponding feature matrix X_k , of size $|S_k| \times (p+1)$, which is composed of the feature vectors x of the $|S_k|$ observations and the target vector Y_k of size $|S_k| \times 1$ that contains 1 for the non-censored observations with $v_i = v^{(k)}$ and 0 otherwise. The coefficients are therefore estimated using a standard regularised least-squares regression method:

$$\beta^{(k)} = (X_k' X_k + \lambda I)^{-1} X_k' Y_k \tag{8}$$

The coefficients for the cumulative hazard rate can be estimated as:

$$B^{(k)} = \sum_{j \ge k} \beta^{(j)} \tag{9}$$

Then, given an observation with feature vector x, its cumulative hazard rate at level $v^{(k)}$ is given by:

$$\Lambda(v^{(k)}|x) = x.B^{(k)} = B_0^{(k)} + B_1^{(k)} x_1 + \ldots + B_p^{(k)} x_p \tag{10}$$

The CDF of the variable V with censored observations can finally be obtained as:

$$\Phi(v^{(k)}|x) = \exp(-\Lambda(v^{(k)}|x)) \tag{11}$$

3.5.2 Using Aalen's regression model to estimate first and second bids' distribution. Let's now consider how Aalen's regression model could be used for solving the issue of left-censored observations in the reserve price optimization problem. In a nutshell, two Aalen's regression models will continuously and adaptively provide the engine with an estimate of the distribution of the first and the second bid distributions independently. At any moment, the estimation of both bid probability distributions can be used in order to estimate the expected revenue for different values of the reserve price. Once again, the models are not parametric, in the sense that they do not assume any prior distribution, unlike several state-of-the-art approaches that assume a log-normal distribution for bids. So, we work on a discretized bid space, using K' bins (or levels): $(b^{(1)}, b^{(2)}, \ldots, b^{(K')})$. Even if this is not required, we will assume for simplicity that $K = K'$ and that the set of discretized values for b_1 and b_2 is the same as for the reserve price (f). Basically, the Aalen's method provides an estimate of the CDF at each of these values. Remarkably, the way the engine maintains probability distributions over the bids is very similar to the way the engine is maintaining a revenue estimation for the different levels of floor: the updates are done for the different discretized levels using a decomposition into latent factors related either to the user or to the placement (but not to the particular $< user, placement >$ pair, avoiding some severe sparsity issues). Moreover, the update equations have actually the same form.

When applying the Aalen's additive model to estimate the first bid distribution, we will assume a latent factor model of the following form: for $k \in [\![1, K]\!]$, and for an auction for an internet user u and an ad placement p

$$\lambda_1^{(k)} = (M_u^{(k)})' N_p^{(k)} + \eta^{(k)} \tag{12}$$

where $\lambda_1^{(k)}$ is the hazard rate of the first bid distribution at level k (bins $b^{(k)}$); $M_u^{(k)}$ and $N_p^{(k)}$ are latent factors columns (of size L) associated respectively to user u and placement p for first bid level $b^{(k)}$; $\eta^{(k)}$ is the decomposition error term at first bid level $b^{(k)}$ and is assumed to be gaussian with zero mean and variance σ_1^2.

All the update equations that will be described hereafter for the first bid distribution are identical for the second bid, except for some details which we will mention explicitly.

As we did for the revenue latent factor estimation, we introduce, for $k \in [\![1, K]\!]$, the following loss function for the estimation problem over $[0, T]$:

$$\begin{aligned} L(\hat{M}^{(k)}, \hat{N}^{(k)}) = &\sum_{a \in \mathcal{D}_T^{(k)}} \gamma_1^{(T-t_a)} (C_a^{(k)} - (\hat{M}_{u_a}^{(k)})' \hat{N}_{p_a}^{(k)})^2 \\ &+ \sum_{u \in \mathbb{U}} \|\hat{M}_u{}^{(k)} - M_0^{(k)}\|_{\Omega_1}^2 + \sum_{p \in \mathbb{P}} \|\hat{N}_p{}^{(k)} - N_0^{(k)}\|_{\Gamma_1}^2 \\ &+ \|\eta^{(k)} - \eta_0^{(k)}\|^2 \end{aligned} \tag{13}$$

where:

- $\mathcal{D}_T^{(k)}$ is the set of auctions up to time T whose first bid or its left-censored value is smaller or equal to level $b^{(k)}$; in other words, this is the set of historical auctions for which $\max(b_1, f_a) \le b^{(k)}$ (f_a is the floor for auction a);
- $C_a^{(k)} = 1$ if the first bid is uncensored AND if the bid belongs to the bin $b^{(k)}$; $C_a^{(k)} = 0$ otherwise;
- $\hat{M}^{(k)}$ and $\hat{N}^{(k)}$ are the matrices of estimated first bid latent factors for users and placements respectively (of size $U \times L$ and $P \times L$);
- γ_1 is a forgetting factor ($\gamma_1 < 1$);
- The last two terms are regularization terms and have a direct bayesian interpretation (section 3.4).

Note that, if we consider the second bid distribution, these definitions should be adapted in the following way:

- $\mathcal{D}_T^{(k)}$ is the set of auctions up to time T whose second bid or its left-censored value is smaller or equal to level $b^{(k)}$; in other words, this is the set of historical auctions for which $\max(b_2, f_a) \le b^{(k)}$ (f_a is the floor for auction a);
- $C_a^{(k)} = 1$ if the second bid is uncensored and if this bid belongs to the bin $b^{(k)}$; $C_a^{(k)} = 0$ otherwise;

As before, the latent factor estimation is performed using an alternating least-squares method. We directly give the update equations for the on-line setting, as they constitute the core of the proposed methods.

So, in order to estimate the latent factors on-line, we need to keep in memory the following terms for each user u, placement p and bid level k :

- $M_{u,cov}^{(k)}$, the time-weighted "covariance" matrix at time T: $M_{u,cov}^{(k)} \doteq \sum_{a \in \mathcal{D}_T^{u,(k)}} \gamma^{(T-t_a)} \hat{N}_{p_a}^{(k)} \hat{N}_{p_a}^{(k)'}$. We define symmetrically $N_{p,cov}^{(k)}$. Note the difference with respect to equation 5: the sum is over the set $\mathcal{D}_T^{u,(k)}$, meaning the set of all historical auctions where user u appeared and for which $\max(b_1, f_a) \le b^{(k)}$;
- $M_{u,obs}^{(k)}$, which represents the current estimate at time T of $\sum_{a \in \mathcal{D}_T^{u,(k)}} \gamma^{(T-t_a)} (C_a^{(k)} - M_0^{(k)'} \hat{N}_{p_a}^{(k)}) \hat{N}_{p_a}^{(k)}$. We define symmetrically $N_{p,obs}^{(k)}$.

With these quantities, the following update equations, after observing the outcome of auction a, are iterated until convergence and for each bid level k:

$$\begin{aligned} \hat{M}_{u_a}^{(k)} &= M_0^{(k)} + \left(\gamma_1^{\Delta t_{u_a}^{(k)}} M_{u_a,cov}^{(k)} + \hat{N}_{p_a}^{(k)} \hat{N}_{p_a}^{(k)'} + \Omega_1^{-1}\right)^{-1} \\ &\quad \left(\gamma_1^{\Delta t_{u_a}} M_{u_a,obs}^{(k)} + (C_a^{(k)} - M_0^{(k)'} \hat{N}_{p_a}^{(k)}) \hat{N}_{p_a}^{(k)}\right) \\ \hat{N}_{p_a}^{(k)} &= N_0^{(k)} + \left(\gamma_1^{\Delta t_{p_a}^{(k)}} N_{p_a,cov}^{(k)} + \hat{M}_{u_a}^{(k)} \hat{M}_{u_a}^{(k)'} + \Gamma_1^{-1}\right)^{-1} \\ &\quad \left(\gamma_1^{\Delta t_{p_a}} N_{p_a,obs}^{(k)} + (C_a^{(k)} - N_0^{(k)'} \hat{M}_{u_a}^{(k)}) \hat{M}_{u_a}^{(k)}\right) \end{aligned} \tag{14}$$

where $\Delta t_{u_a}^{(k)}$ and $\Delta t_{p_a}^{(k)}$ are respectively the time intervals since the last impression for user u_a and the last impression for placement p_a, restricted to past auctions for which $\max(b_1, f_a) \le b^{(k)}$. In practice, one or two iterations are sufficient.

Finally, the following update formulae are applied:

$$\begin{aligned} M_{u_a,cov}^{(k)} &\leftarrow \gamma_1^{\Delta t_{u_a}^{(k)}} M_{u_a,cov}^{(k)} + \hat{N}_{p_a}^{(k)} \hat{N}_{p_a}^{(k)'} \\ M_{u_a,obs}^{(k)} &\leftarrow \gamma_1^{\Delta t_{u_a}^{(k)}} M_{u_a,obs}^{(k)} + (C_a^{(k)} - M_0^{(k)'} \hat{N}_{p_a}^{(k)}) \hat{N}_{p_a}^{(k)} \\ N_{p_a,cov}^{(k)} &\leftarrow \gamma_1^{\Delta t_{p_a}^{(k)}} \times N_{p_a,cov}^{(k)} + \hat{M}_{u_a}^{(k)} \hat{M}_{u_a}^{(k)'} \\ N_{p_a,obs}^{(k)} &\leftarrow \gamma_1^{\Delta t_{p_a}^{(k)}} \times N_{p_a,obs}^{(k)} + (C_a^{(k)} - N_0^{(k)'} \hat{M}_{u_a}^{(k)}) \hat{M}_{u_a}^{(k)} \end{aligned} \tag{15}$$

3.5.3 Revenue profile estimation from the estimated bid distributions. We can derive the CDF of b_1 and b_2, from their hazard rates: for instance, the CDF of $b1$, denoted by $\Phi_1(b^{(k)})$ can be computed as: $\Phi_1(b^{(k)}|u, p) = \exp(-\sum_{j \ge k} \lambda_1^{(j)}(u, p))$. The CDF of b_1 and b_2 can then be used to estimate the expected revenue at each discrete value of the floor price and this estimate is then used to feed the Revenue Profile Modeller.

Let's first consider the full censorship case ($b_2 \leq b_1 \leq f_a$ where f_a is the floor selected for the current auction a). Since the revenue is given by $R(f, b_1, b_2) = b_2 \mathbb{1}_{b_2 > f} + f \mathbb{1}_{b_2 \leq f \leq b_1}$, the expected revenue in the continuous case can be estimated as:

$$\begin{aligned}
&\mathbb{E}(R(f, b_1, b_2 | b_2 \leq b_1 \leq f_a)) = \\
&\int_f^{f_a} P(b_2 > t | b_2 \leq b_1 \leq f_a) dt + f P(f \leq b_1 | b_2 \leq b_1 \leq f_a) \\
&\simeq \int_f^{f_a} P(b_2 > t | b_2 \leq f_a) dt + f P(f \leq b_1 | b_2 \leq b_1 \leq f_a)
\end{aligned} \tag{16}$$

Details of the derivation can be found in equation (2) of [6]. The second equation emphasizes the approximation that allows us to model independently the first and second distributions, instead of modelling the joint distributions. Experimentally, this approximation could be shown to have virtually no impact on the computation of the expected revenue, at least with the auction datasets we used.

This second equation can be converted for the discrete case into the following equation:

$$\begin{aligned}
&\forall f^{(k)} < f_a, \quad \mathbb{E}_{b_1, b_2}(R(f^{(k)}, b_1, b_2 | b_2 \leq b_1 \leq f_a)) = \\
&\sum_{k'=k}^{f_a} f^{(k')} \tilde{\phi_2}(b^{(k')}) - f^{(k)}.(1 - \tilde{\Phi_2}(f^{(k)})) + f^{(k)}.(1 - \tilde{\Phi_1}(f^{(k)}))
\end{aligned} \tag{17}$$

where $\tilde{\Phi_1}$ and $\tilde{\Phi_2}$ are the re-normalised c.d.f.'s of the first and second bid respectively, conditioned by the fact that they should be smaller than f_a: $\tilde{\Phi_1}(f) = \frac{\Phi_1(f)}{\Phi_1(f_a)}$, $\tilde{\Phi_2}(f) = \frac{\Phi_2(f)}{\Phi_2(f_a)}$ with Φ_1 and Φ_2 the (non-conditional) c.d.f.'s of the first and second bid respectively, while $\tilde{\phi_2}$ is the (discrete) renormalised p.d.f of the second bid corresponding to the $\tilde{\Phi_2}$ CDF. For levels $f^{(k)} \geq f_a$, the revenue is equal to 0.

For the half-censored case (b_1 observed and $f_a \leq b_1$), this formula becomes:

$$\begin{aligned}
&\mathbb{E}_{b_2}(R(f^{(k)}, b_1, b_2 | b_2 \leq f_a)) = \\
&\sum_{k'=k}^{f_a} f^{(k')} \tilde{\phi_2}(b^{(k')}) - f^{(k)}.(1 - \tilde{\Phi_2}(f^{(k)})) + f^{(k)} \quad \forall f^{(k)} < f_a
\end{aligned} \tag{18}$$

For levels $f^{(k)} \geq f_a$, the revenue is not "censored" and is equal to $f^{(k)}$ if $f^{(k)} \leq b_1$ and 0 otherwise.

3.6 Introduction of additional features

We detail briefly how to introduce contextual features other than the user's and placement's identifier into the model , e.g. the time of the day or the user's device. Let's θ be the corresponding feature vectors. For the sake of simplicity, we assume that the dependency between the feature vector and the revenue is linear and that it simply combines additively with the biases and latent factors:

$$R^{(k)} = \beta^{(k)} + (X_u^{(k)})' Y_p^{(k)} + \theta' Z^{(k)} + \epsilon^{(k)} \tag{19}$$

In this setting, the formulae to estimate $\beta^{(k)}$ and the latent factors $X_u^{(k)}$ and $Y_p^{(k)}$ are kept unchanged, except that $R^{(k)}$ is replaced by $(R^{(k)} - \theta' Z^{(k)})$. It is straightforward to derive the formulae to estimate adaptively the parameters $Z^{(k)}$: it is similar to the update equations of $\beta^{(k)}$, except that it uses θ as regressor, instead of the **1** constant.

3.7 Computational complexity of the engine

Let K be the number of reserve price levels, L be the size of the latent factors space and I be the number of iterations performed when updating the latent factors.

The most computationally expensive operations are the updates of the latent factors used in the revenue profile and bid distribution modellers. These updates require a $L \times L$ matrix inversion for each iteration and for each reserve price level, i.e $K \times I$ matrix inversions.

In practice, an appropriate value for K is around 100, and 5 for I. L is also low (see section 4), which makes the engine suitable for real-world applications.

4 RESULTS

In this section we describe results obtained from comparing our method with baseline approaches that are appropriate for the task and also with state-of-the-art methods that have been proposed recently in the literature.

The evaluation is performed on a real advertising publisher dataset containing more than 4.4 millions of auctions data collected over one week. It contains 367K unique users and 2K unique placements, as well as the first and second bid values (non-censored) of each auction which provides the ground truth to evaluate our method with respect to the full information (non-censored) setting.

The dataset is divided into two subsets: the training set (containing the first 3 days of observations, approximately 3/7 of the total number of observations) and the test set. The results given below are estimated on the test set.

For all the evaluations, the metric to measure peformance is the resulting revenue. It is a reasonable measure since the primary goal is to maximize the revenue, conversely to other works [8] which try to predict the highest bid in which case the estimation error is a more appropriate metric.

Initially, the revenue achieved when applying the reserve price predicted by the model is compared to the following two reserve price setting baselines:

- *NO_RES*: reserve price equal to 0;
- *PL_RES*: setting an optimal reserve price per placement determined on the training set (using uncensored bid values);
- *PL_RES_ONLINE*: setting an optimal reserve price per placement which is estimated online as the reserve price maximizing an exponentially-weighted moving average of the revenue (using uncensored bid values)

We consider 3 settings:

S1: The ideal one (or easy one), where the training is uncensored (but the test set is censored); at the end of the training phase, as usual, the identified values of the latent factors are used as initial values when starting the test phase;

S2: The hard one, where both the training and test sets are censored; note that the censorship of the training set is fixed to the historical prices and could not be modified, while the censorship level in the test set could be controlled by the reserve price optimisation strategy (we use the knowledge of both b_1 and b_2 to simulate the revenue of the proposed strategies); moreover, we assume that the training set could be used only as a "development set" to tune the hyper-parameters, but not to initialise the values of the latent factors (as if a reset operation has been applied just before the test set); the goal of this constraint is to emphasize the "cold-start" and adaptive capabilities of our algorithm;

S3: An intermediate one, where only the test set is censored; but here also, we assume that the training set could be used only as a "development set" to tune the hyper-parameters, but not to initialise the values of the latent factors.

Besides the two baselines, we consider 4 variants of our method:

M1: The complete one, using both the revenue profile modeller and the bid distribution modeller when the bid information is censored

M2: A variant where only the revenue profile modeller is used and, to handle the censorship issue, we arbitrarily fix b_1 and b_2 to 0 in case of full censorship, and b_2 equal to f_a in case of half-censorship; intuitively, this amounts to never favouring "higher" levels of floor in case of censorship so that we can promote exploration of low levels of floor (remember that, when we know the revenue for a floor, we automatically know the revenue for all floors that are larger than this floor). So, this is an indirect way of controlling the exploration/exploitation trade-off;

M3: A variant where only the revenue profile modeller is used and, to handle the censorship issue, we simply skip it, meaning that we do not update the revenue profile for the part which is unknown due to censorship (but we update it for all levels that are larger or equal to the current reserve price);

M4: the same variant as **M3**, but instead of selecting the reserve price whose predicted expected revenue is the largest, we consider the Lin-UCB selection strategy a term directly takes its inspiration from the *Lin-UCB* sampling strategy for contextual bandits ([5]): basically, it amounts to add an extra term to the predicted revenue; this term represents some kind of upper confidence interval on the predicted revenue, taking into account the current uncertainties over the latent factors; it allows us to control the trade-off between exploration and exploitation, as higher floor levels could lead to censored observations and "weaker" updates. Note that **M1** avoids making such a compromise, by continually estimating the missing information.

Only (**M1**) uses the bid distribution modeller component.

In the update equations of the adaptive methods, the parameters (biases and latent factors) were initialised to small random values (gaussian with 0.1 variance).

The performances of the different methods, namely the average revenues on the Test Set (in arbitrary monetary units) are given in the following table:

	S1	S2	S3
NO_RES	2.5978	N/A	N/A
PL_RES	3.6222	N/A	N/A
PL_RES_ONLINE	3.7154	N/A	N/A
M1	4.0663	3.9955	3.9957
M2	3.9012	3.7948	3.7936
M3	3.8954	3.7369	3.7468
M4	3.9306	3.821	3.8209
Oracle (knowing b_1)		8.6552	

Some remarks about the results and the choice of the hyper-parameters:

- Starting the model from scratch at the beginning of the test set finally has a small impact on the performance; this is not surprising, knowing that there are a lot of "flash" users (i.e. new users who never appeared before and will disappear a few minutes later) for which solving the cold-start problem is crucial;
- A censored training set is not detrimental to performance: results are nearly the same than with a uncensored training set;
- In the case of both training and test sets uncensored (so that no censorship-dealing strategy should be used), the revenue profile modeller gives a performance of 4.163 (4.1294 if we apply a "reset" operation before the test set, to emphasize cold-start performance). The method **M1** is relatively close to this level of performance and it shows that the bid distribution modeller performs well;
- The hyper-parameters are determined using grid-search (6 discrete values per hyper-parameter on a logarithmic scale, from 10^{-6} to 10^{-1} for the forgetting factors and from 10^{0} to 10^{5} for the constant diagonal covariance priors), focusing on the ones that give the best average revenue on the training set. Optimal results are reached when the observations are forgotten after a few minutes for users and a few hours for placements;
- The dimension of the latent space can be kept low (in our case 2): once the latent factors corresponding to user and placement biases are added in the model, adding new latent factors does not improve very significantly the results to the price of a high complexity. This comes probably from the relatively low number of observations per user, which improves the risk of over-fitting if the dimension is too large. The optimal value for the latent space dimension is probably dependent on the dataset

Finally, we have compared our method with three state-of-the-art approaches: the one based on a Bayesian smoothing of the revenue function (the parametric approach based on an *EM*-like algorithm as described in [7]), the one based on an assumed log-normal distribution of the first bid [9] and the one based on the simple adaptation mechanism ("increase the floor when it is below the first bid; decrease it when it is larger"), as described in [9] but extended to maintain one optimal floor value per placement. Note that the former method is not adaptive (the parameters are fixed after a training phase), while the last two methods are. Both methods assume that the bid information is uncensored. Therefore, in order to keep the comparison fair, we compared these benchmark methods with ours in the full information setting (training and test sets uncensored). The average revenues on the test set are summarized in the following table:

	S1
Bayesian Smoothing (non-adaptive)	2.893
Log-Normal Bid Distribution (adaptive)	2.932
Simple Bi-directional Inc/Dec (adaptive)	3.627
M1	4.163

The relatively poor performance of our benchmark methods most probably comes from the fact that their underlying assumptions are violated in practice. Indeed, as far as method [9] is concerned, assuming a log-normal distribution for the first bid and taking the mean minus a small constant for setting the reserve price is a too simplistic strategy, due to the high skewness of the revenue function as we explained before. For the Bayesian smoothing method of [7], considering that the optimal reserve price could be expressed as a linear function of the available features – which means here that it could be expressed as the sum of a user weight and a placement weight – seems also to be too simplistic. Note that this method uses only $P + U$ parameters (when only information available is the user identifier and the placement identifier), while our method uses $K.(P + U)$ parameters; moreover, this method does not take into account the time-varying aspects of the problem and this also explains its low performance. The simple adaptive "increase/decrease" method turned out to be only slightly superior to the non-adaptive *PL_RES* strategy, implying that the adaptation mechanism did not really succeed in capturing the time-varying properties of the first bid distribution.

We have also measured the practical efficiency of the proposed method in terms of CPU response time using a standard computer (μP 3.50GHz, RAM 8GB). Updating the full model(M1 above) and estimating the optimal floor price in the worst case (under full censorship) requires only 0.38 ms, and even less without censorship (0.21 ms). This is below the limit of 1ms at which the optimal floor must be decided.

5 CONCLUSIONS AND FUTURE DIRECTIONS

Deploying a revenue maximization engine through the optimal setting of the reserve price for real-time-bidding auctions remains a challenging industrial problem: the "very short latency" constraint with its implication on algorithm complexity, the sparsity of user and placement information and, last but not least, the highly time-varying environment all raise important issues to be solved.

In the deployed engine, we adopted a non-parametric approach to adaptively predict, by an on-line matrix factorisation approach, the revenue for each floor level as a function of the user and the ad placement. To compensate for the lack of full bid information, which is due to the intrinsic censorship of the auction mechanism, we used a similar approach for predicting the first and second bid distributions through the Aalen's additive regression model. This approach was – and is still – validated by the revenue increase it brings to hundreds of publishers. The experimental results showed that the proposed approach outperforms state-of-the-art methods and, in particular, that the average revenue almost reaches the level of the non-censored setting.

The model can be extended to deal with more complex non-linear interactions than the one assumed by a simple matrix factorization, especially when some extra user or placement features are available, keeping in mind that the real-time constraints could put some restriction on the class of models in practice.

Finally, although the parameters of the engine are time-adaptive, the impact of the dynamic reserve prices on the advertisers' bidding behavior is not modeled explicitly. A more global game-theoretic approach should be introduced to tackle this issue, but this introduces some closed loops that would make the problem much harder to solve.

Aknowledgment: This work was partially funded by the French Government under the grant <ANR-13-CORD-0020> (ALICIA Project).

REFERENCES

[1] O. O. Aalen. A linear regression model for the analysis of life times. *Statistics in Medicine*, 8(8):907–925, 1989.

[2] N. Cesa-Bianchi, C. Gentile, and Y. Mansour. Regret minimization for reserve prices in second-price auctions. *IEEE Trans. Information Theory*, 61(1):549–564, 2015.

[3] Y. Cui, R. Zhang, W. Li, and J. Mao. Bid landscape forecasting in online ad exchange marketplace. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, 2011.

[4] E. Ikonomovska, S. Jafarpour, and A. Dasdan. Real-time bid prediction using thompson sampling-based expert selection. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, 2015.

[5] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, 2010.

[6] M. Mohri and A. M. Medina. Learning theory and algorithms for revenue optimization in second price auctions with reserve. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 262–270, 2014.

[7] M. R. Rudolph, J. G. Ellis, and D. M. Blei. Objective variables for probabilistic revenue maximization in second-price auctions with reserve. In *International Conference on World Wide Web*, WWW '16, 2016.

[8] W. C.-H. Wu, M.-Y. Yeh, and M.-S. Chen. Predicting winning price in real time bidding with censored data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, 2015.

[9] S. Yuan, J. Wang, B. Chen, P. Mason, and S. Seljan. An empirical study of reserve price optimisation in real-time bidding. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, 2014.

[10] S. Yuan, J. Wang, and X. Zhao. Real-time bidding for online advertising: Measurement and analysis. In *International Workshop on Data Mining for Online Advertising*, ADKDD '13, 2013.