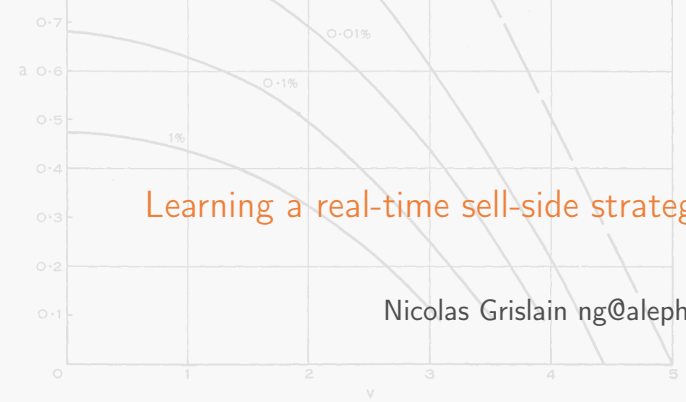


Fig. 2.—Regions of usefulness of Gauss-Herr[illegible] for given error limits. The dashed lines $v=5-2a$, $a=1\cdot2$, separate the regions in which different methods were used for the computation of Table 1.

Learning a real-time sell-side strategy on RTB platforms

Nicolas Grislain ng@alephd.com

October 2014

Introduction

- We study auctions on which ad spaces are sold by publishers to advertisers in milliseconds.
- We focus on optimizing the gain of publishers, based on real-time floor strategies.
- After a floor has been chosen, and the auction happened, the payoff — had a different floor been set — is partially known.

3 main difficulties

1. Learn a joint bid (first and second) distribution based on partial observations
2. Make one's strategy depend on a sparse context ($user \times tag$)
3. Set a floor to exploit / explore

AlephD in the RTB process

- A user visits a page
- An ad-call is sent to the Ad-exchange
- A bid request is sent to the buyers
- The highest bidder wins and get to display a creative to the user

The whole process lasts $\sim 100ms$

AlephD in the RTB process

- A user visits a page
- An ad-call is sent to the Ad-exchange
- **A *data call* is sent to AlephD**
- **Data — a floor price — is sent by AlephD to the Ad-exchange in 10 ms**
- A bid request is sent to the buyers
- The highest bidder wins and get to display a creative to the user
- **A *pixel* with data about the auction is sent to AlephD**

The whole process lasts $\sim 100ms$

The setting

- $\mathcal{A} = \{a_1, \ldots, a_A\}$ the set of successive auctions, happening at time $\{t_{a_1}, \ldots, t_{a_A}\}$
- $\mathcal{B} = \{b_{a,1} > \ldots > b_{a,B}\}$ the set of ordered bids for auction a — bidders do correlate
- u_a and g_a the user and the tag of a
- The payoff as a function of r_a

$$\pi_a(b_{a,1}, b_{a,2}, r_a) = \begin{cases} b_{a,2} & \text{if } b_{a,2} \geq r_a \\ r_a & \text{if } b_{a,1} \geq r_a > b_{a,2} \\ 0 & \text{if } r_a > b_{a,1} \end{cases}$$

$$= \max(b_{a,2}, r_a)\, \mathbf{1}_{r_a < b_{a,1}}$$

The setting

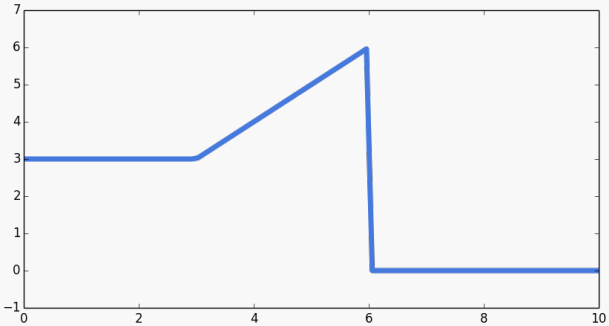
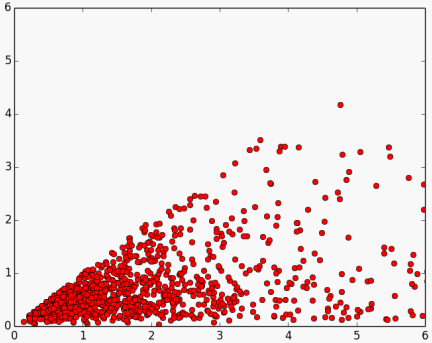
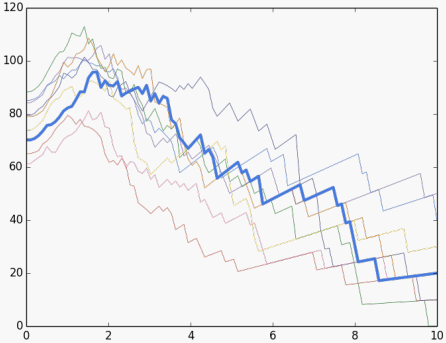


Figure : Payoff Structure of a Second-Price Ad-Auction

The setting

Conditional on the available information: the joint distribution of (b_1, b_2) leads to the expected payoff





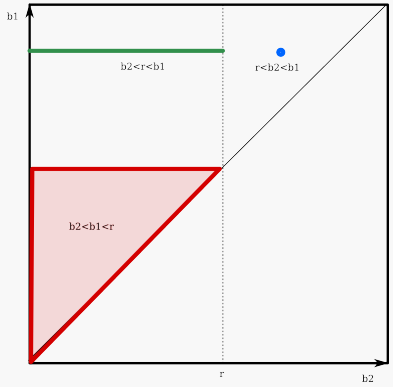
The setting

The result of the auction a is imperfectly observed.

$$o_a = \begin{cases} (a, u_a, g_a, r_a, b_{a,1}, b_{a,2}) & \text{if } b_{a,2} \geq r_a \\ (a, u_a, g_a, r_a, b_{a,1}, r_a) & \text{if } b_{a,1} \geq r_a > b_{a,2} \\ (a, u_a, g_a, r_a, 0, 0) & \text{if } r_a > b_{a,1} \end{cases}$$

- Hence the difficulty of estimating the joint distribution of $(b_{a,1}, b_{a,2})$ needed for the estimation of the payoff $\pi_a(b_{a,1}, b_{a,2}, r_a) = \max(b_{a,2}, r_a)\,\mathbf{1}_{r_a < b_{a,1}}$
- Hence the need to explore / exploit

The setting



First challenge: learn the $(b_{a,1}, b_{a,2})$ distribution in various contexts

Different approaches were tested:

- Make some simple assumptions on $(b_{a,1}, b_{a,2})$ when not observed and model directly the payoff
- Learn the $(b_{a,1}, b_{a,2})$ distribution using a semi-parametric model
- Learn the $(b_{a,1}, b_{a,2})$ distribution using a RBM

sparse environment

There is a case for keeping the dimension of the model low.

First challenge: some simple assumptions on $(b_{a,1}, b_{a,2})$

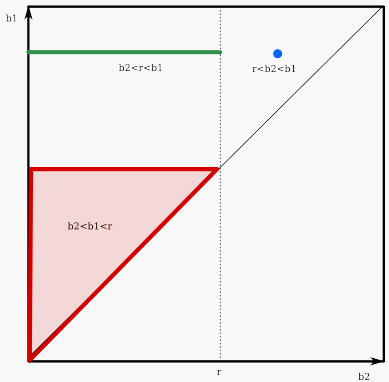
- $b_{a,2} = \alpha b_{a,1}^{\beta}$ when not observed
- $b_{a,2} = b_{a,1} = 0$ when not observed

The first assumption is not very frequently used and not so false, but the more aggressive the strategy the more often it is used.
$\Rightarrow$ The exploitation reduces the accuracy of our approach.
The second assumption is very false but rarely observed and implement some exploring behaviour.

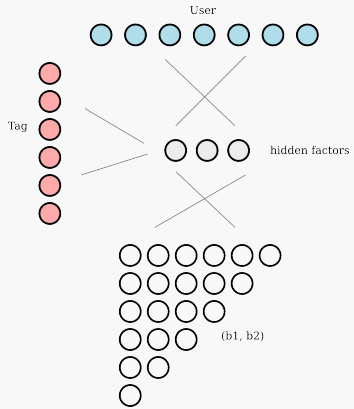
First challenge: a semi-parametric model

$$f(f_k, b_2) = \frac{e^{\alpha_k}}{I(\alpha., \beta.)}(f_k - b_2)^{\beta_k - 1}\mathbf{1}_{[0,f_k]}(b_2)$$

Whenever $b_1 \in [f_k, f_{k+1}[$
The log-likelihood is computed according to one of the three cases of information availability.



First challenge: the RBM approach



- The *triangle* part is interpreted as a multinomial logistic model.
- *User* and *tag* units are binary.
- A limited number of hidden units forces the dimension down which tackles sparcity

Second challenge: cope with sparsity

N users $\times$ K placements. Let's assume (u_i, g_j) summarize the context. We want to compute an optimal floor. Different approaches were tested with a simple model of payoff

- The naïve approach of learning an adaptative value for each (u_i, g_j) pair with SGD
- An adaptative NMF approach
- Learn the $(b_{a,1}, b_{a,2})$ distribution using a RBM

sparse environment

There is a case for keeping the dimension of the model low.

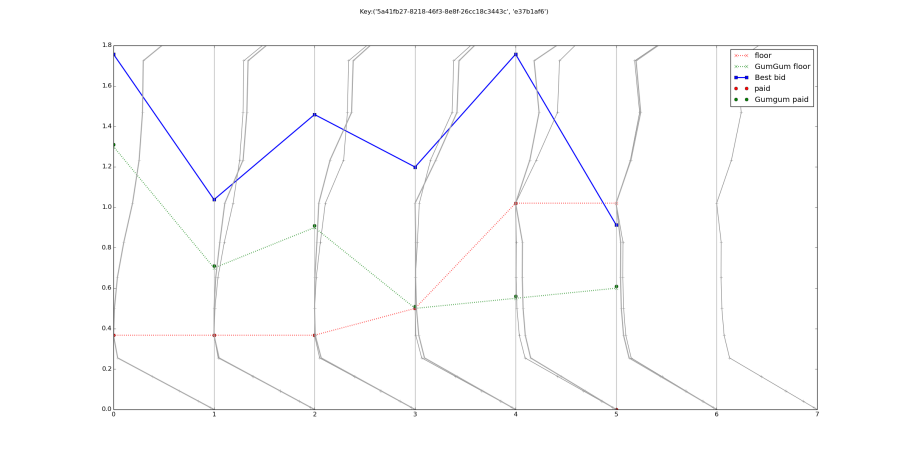
Second challenge: the naïve method works in practice

- The most naïve approach: use the last observed bid for a given (*user* $\times$ *tag*) or *user*. It works (The user x tag information is very relevant) and is already a technical challenge.
- Update an exponentially weighted moving average of expected payoff for each level of floor.

It is already pretty efficient — +30% revenue on some websites

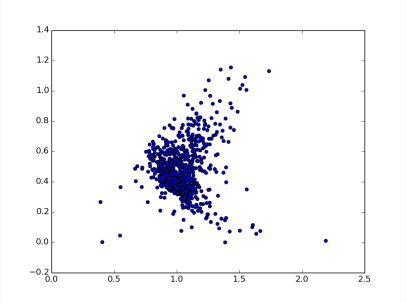
But not in all the cases — especially when user do not come back very often or when the number of tags is large

Second challenge: the naïve method works in practice

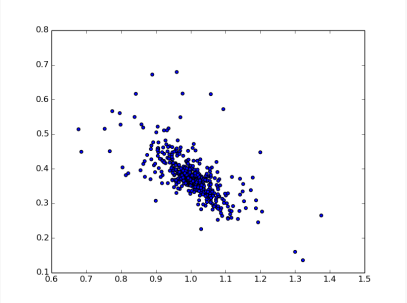


Second challenge: an adaptative NMF approach

- By using NMF we learn about users, and tags
- We reduce the number of parameters from $N \times K$ to $N \times r + K \times r$



Tags



Users

Third challenge: explore / exploit

How to adapt bandits ideas to our case ? Different approaches were or will be tested:

- The naïve approach of playing 0 sometimes to have the full information.
- A randomization scheme based on the UCB idea

sparse environment

Today we use heuristics vaguely inspired from UCB ideas.

Thank you

And by the way, we are recruiting...

...and we have a lot of data to play with (including highest bids)