Pacing algorithm

Thomas Maurice

September 7, 2020

Table of contents

Digital Out-Of-Home

- Digital Out-Of-Home (DOOH) advertising is a marketing channel where an ad is dynamically and digitally displayed in outdoor public spaces.

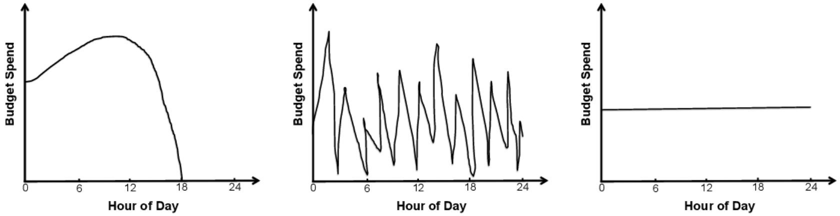
Figure: Advertising in a mall

Displayce in DOOH

- Main intermediary between supply (publishers) and demand (advertisers)
- Programmatic purchasing platform
  1. Booking
  2. RTB

Smooth budget delivery

- In open RTB: Displayce buys impressions in real time
- For each bid request, the bidder chooses whether to buy or not
- Importance of a pacing algorithm to smooth the budget expenditure



(a) Premature Stop (b) Fluctuating Budget (c) Uniform Pacing

Figure: Examples of expenditures

Table of contents

Data

- To build the algorithm, we will use a data set of the following form:

	ID	Imps	Price	Win	Seconds notif
2020-07-08 06:00:00	1.0	10.0	10.0	True	365.0
2020-07-08 06:00:00	2.0	1.0	1.0	True	361.0
2020-07-08 06:00:01	3.0	5.0	5.0	True	555.0
2020-07-08 06:00:01	4.0	1.0	1.0	True	645.0
2020-07-08 06:00:03	5.0	1.0	1.0	True	310.0
2020-07-08 06:00:03	6.0	2.0	2.0	True	355.0

Table of contents

Hourly capping

- The simplest algorithm is a hourly capping
- Cons:
  1. All the budget is spent at the beginning of the hour

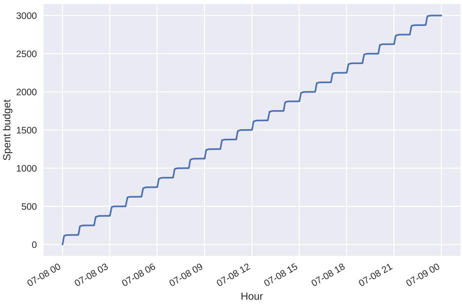


Figure: Hourly capping

Hourly capping

- The simplest algorithm is a hourly capping
- Cons:
  1. All the budget is spent at the beginning of the hour
  2. No catch-up possible

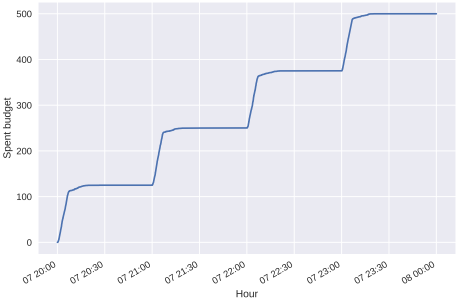


Figure: Hourly capping

Table of contents

Budget per second

- Calculate a budget per second to have a uniform expenditure over the day
- Budget per second is recalculated each time a bid request is received
- We can calculate the budget per second noted b_t as follow:

$$b_{t+1} = \left(B - \sum_{s=1}^{t} S(s) \right) \frac{1}{T - t}$$

Where b_{t+1} is the budget to be allocated to the second $t+1$, B is the total budget for the day, $S(s)$ is the actual expenditure to the second t and finally $T-t$ is the estimated seconds until the end of the last bid request.

Improvement

- We need to determine if there is a slowdown in the frequency of reception of impressions
- The budget per second can be improved by taking into account its against time variation. If b_t increases, it means that less and less impressions are bought
- We can calculate the variation $v_t = b_t - b_{t-1}$ and the speed of the variation $a_t = v_t - v_{t-1}$. We denote $\bar{v}$ and $\bar{a}$ respectively the average variation of b_t over the last 30 minutes and the average speed in the variation of b_t
- The calculation of b_t becomes:

$$b_{t+1} = \left(B - \sum_{s=1}^{t} S(s)\right) \frac{1 + (\bar{a}\bar{v})}{T - t}$$

Uniform pacing

- The expenditure over the day is much smoother and more uniform than the hourly capping

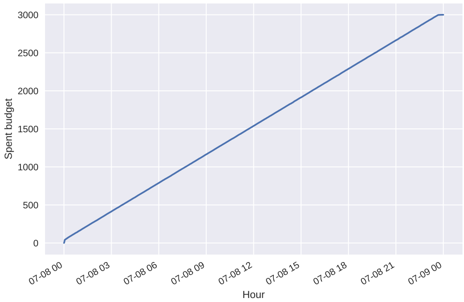


Figure: Per second budget

Catch-up of the expenditure

- The unspent budget from the previous hours is caught up at the end of the day contrary to the hourly capping
- Withdraw: the expenditure is not uniform over the day. The slowdown has to be predicted in order to smooth the expenditure over the day

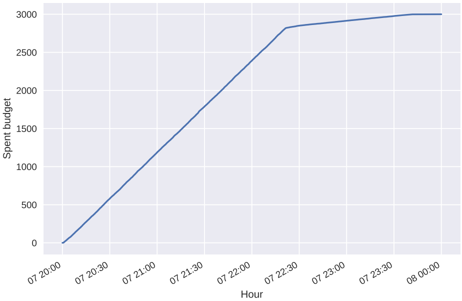


Figure: Per second budget

Evolutive hourly budget

- Predict the proportion of impressions received per hour by using historical data and performing a simple linear regression

$$Prop_i = \alpha + \boldsymbol{\beta_k}\mathbf{X_k} + \varepsilon_i$$

Where $\boldsymbol{\beta_k}$ is the vector of coefficients associated with the vector of explanatory variables.

- The formula of b_t then becomes:

$$b_{t+1} = \left(B(h) - \sum_{s=1}^{t} S(s)\right) \frac{1 + (\bar{a}\bar{v})}{T - t}$$

Where $B(h)$ is the budget allocated to the hour h, $\sum_{s=1}^{t} S(s)$ is the budget spent in the current hour h

Evolutive hourly budget

- The algorithm catches up with the unspent budget much more uniformly than the previous one.

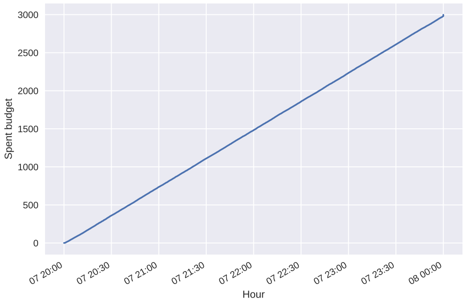


Figure: Evolutive hourly budget

Comparison

- We can compare 3 algorithms over the same dataset. The third algorithm (red curve) catches-up the expenditure and do it uniformly.

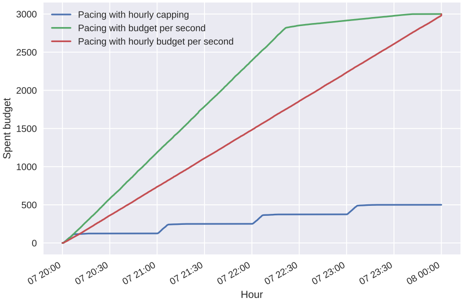


Figure: Expenditure comparison

Table of contents

Meta-class

- Account for the algorithm the possibility to add a spatial dimension to the pacing
- Difficult time zones integration
- Initially allocate a budget per time zone and run the time pacing algorithm independently for each time zone
- Create a meta-class that will create its own instances of the pacing class for each time zone
- Problem: how much budget can we allocate for each time zone?

Dynamic budget reallocation

- Initialization: the algorithm distribute the budget equally among the time zones.
- The objective is to adjust in real time the budget allocated
- Each bid request received, the algorithm calculates the proportion of bid request bought. If it is 100%, we need to reallocate the budget.
- Numerical example: Let T_1, T_2, T_3 and T_4 stand for 4 time zones. Total budget: €1000. At the very beginning: $T_1 = T_2 = T_3 = T_4 = 250$. If T_1 buys 100% and we expect it to buy only half of the original objective: we reallocate 50% of €250 to the other time zones, namely, $\frac{125}{3}$ for each time zone. Then, $T_1 = 125$ and $T_2 = T_3 = T_4 = 250 + \dfrac{125}{3}$

Graph

- The following graph shows the pacing algorithm on 11 time zones
- The algorithm has reallocated the budget between time zones

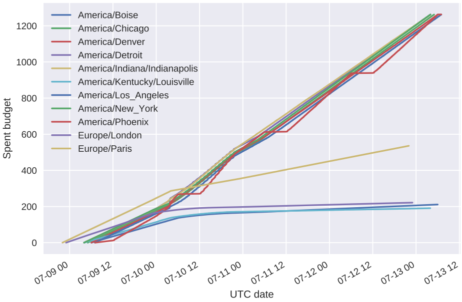


Figure: Expenditure per time zone

Table of contents

Conclusion

- Implementation of a pacing algorithm that better meets the objectives:
- Integration of the possibility to have a pacing between time zones
- What's next?
  1. Improve the performance of the algorithm
  2. Provide an API to integrate the algorithm in production environment
- What I have gained personally:
  1. Progress in research methodology
  2. Improvement of Python skills

END