A DSP view on attribution API

Alois Bissuel, Thomas Prieur, Maxime Vono

Criteo

Objective

- Provide our view on ARA
- Discuss what to improve with Attribution API to ensure good adoption

Measurement, Machine Learning & Budgeting

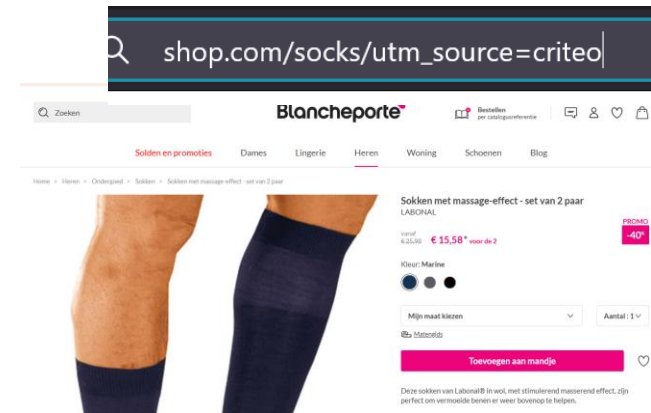- ## Attribution to a specific impression

  ➢ Required for machine learning (learn "good" impression features)

DSP's USP

Attribution to a channel

➢ Not linked to a specific impression

➢ Requires consent on advertiser website

➢ Used to prove value of advertising

➢ Advertiser are ROI centred

➢ Advertiser use Measurement tools aggregating data from all channels (Google Analytics) to assign budget based on performance

Summary of ARA testing findings

✅
- Basic Reporting, PV / PC (campaign level)
- API dedicated for ML (event level)
- Cross-env attribution (Android App <> Chrome)
- Requeryring / reprocessing

❌
- Advanced Reporting (fine level)
- Omnichannel, cross device, etc.

❓
- Performance of ML optimization ?

For more details, check our blog posts :

Attribution API : **pros** & cons

What works

- Simple primitives
- Flexible primitives
  - No constraints on event types

- Clear API structure
- Flexibility on the aggregation method (TEE or MPC)

Attribution API: pros & **cons**

What doesn't work for adoption at scale

- Privacy budget depletion by other adtechs
    - Clear coordination issue
- No ML
    - Though summation could be an acceptable primitive
- No standard for noise management
- No cross-advertiser aggregation (yet!)
    - Long tail of small clients where aggregation per advertiser is not possible
- Limited way of integrating data from the conversion in the histogram

We don't know

- Impact of the "new" user-level privacy granularity on the utility

ML on aggregated data : what we need

We need to be able, at least, to train GLMs (incl. logistic regression):

- Two main approaches we considered:
    1. "Online" learning: gradient is expressed through sums, so sum is a required primitive
    2. Learning from noisy label proportions (LP): requires to build noisy LP first and then standard training with relevant loss function

- We built a prod-ready solution for 1 under Privacy Sandbox Aggregation Service constraints. Not ideal but acceptable performance with epsilon greater than 10

- Need: DP budget constraints do not result in a higher noise, compared to what was possible with Privacy Sandbox Aggregation Service

ML on aggregated data : what we need

With DP (epsilon = 10)
~ 9% downlift

0.35

Without DP

0.32
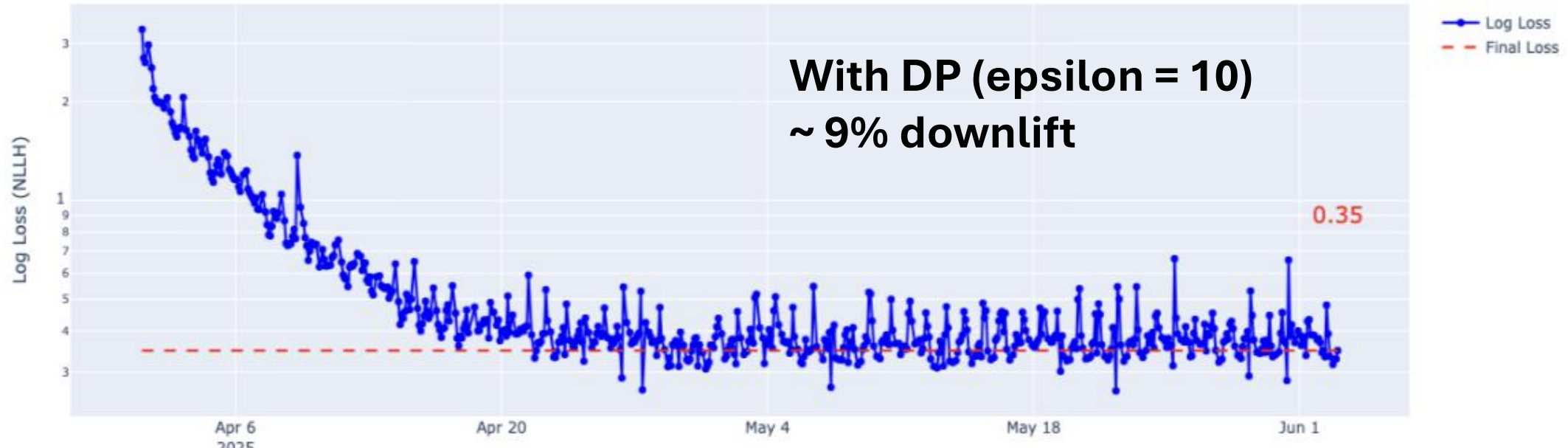
Online gradient descent training loss