Future-Proof Learning - On Learning via Aggregation Servers

Maxime Vono m.vono@criteo.com
Fabian Höring f.horing@criteo.com

Agenda

1. Motivation / Context

2. Learning using (sum/count) aggregation APIs

3. Learning using general aggregation APIs
   (# learning in trusted servers)

Agenda

1. Motivation / Context

2. Learning using (sum/count) aggregation APIs

3. Learning using general aggregation APIs
   (# learning in trusted servers)

Hybrid proposal feature scope

- Hybrid proposal is the right direction to move forward, important to ship something and test
- Supports attributing conversions to impressions and computing aggregated histograms
- Post click attribution in itself can already be handled by statically annotating links
- Using example.com/ad_campaign1 instead of example.com for the ad campaign, the same link is displayed to many users

Campaign optimization - ML training mechanism

Input: features (user, contextual & ad) + conversion label (e.g. click, visit, sales)

Goal: optimize advertising campaigns

How: learn probabilities of (rare) events to build bidding models

Privacy Sandbox CMA test

- Industry-wide Chrome coordinated experiment related to 3rd-party cookie deprecation

- Experiments on 3 Chrome injected populations
  - status quo (with 3rd party cookies)
  - cookieless
  - Privacy Sandbox APIs + cookieless

Privacy Sandbox CMA test

- Strong impact of publisher revenue Status quo vs Cookieless (~ -50%)

- **Life After Cookies: Who Uses Google's Privacy Sandbox? - Garrett Johnson**

Privacy Sandbox CMA test

- Current Hybrid proposal will, if feature implemented, at best allow contextual campaign optimization, e.g. features ad size, publisher domain & measured attributed label

- Contextual campaign optimization might not be sufficient as the down lift mostly comes from missing infos about user & advertiser data

- It is important to include more targeting data into the campaign optimization in a privacy preserving way

Private learning constraints

- Noisy event level reporting
  - **local DP** applied on sensitive feature/labels

- ML training via (sum/count) aggregation API
  - **global DP** applied to aggregated statistics

- ML training via a trusted server (general aggregation API)
  - **global DP** applied to model training (loss, gradient or weights)

Local DP Learning Paradigm

PROS	CONS

- agnostic to the reporting downstream tasks
- noise is added once
- easy to use within current stack

- learning is biased (de-biasing possible with label DP)
- more noise than global DP
- when including the features in addition to label performance is low

Local DP Learning Paradigm – Further steps

- Local DP on dense vectors will likely help (preserving feature semantics)
- Research route – how to "de-bias" user features contamined with local DP noise?
- Research on local DP might help to learn from label proportions

Agenda

1. Motivation / Context
2. Learning using (sum/count/mean) aggregation APIs
3. Learning using general aggregation APIs
   (# learning in trusted servers)

ML training via aggregation API

PROS

- less noise than local DP, in general
- aggregate statistics are partially re-usable across tasks

CONS

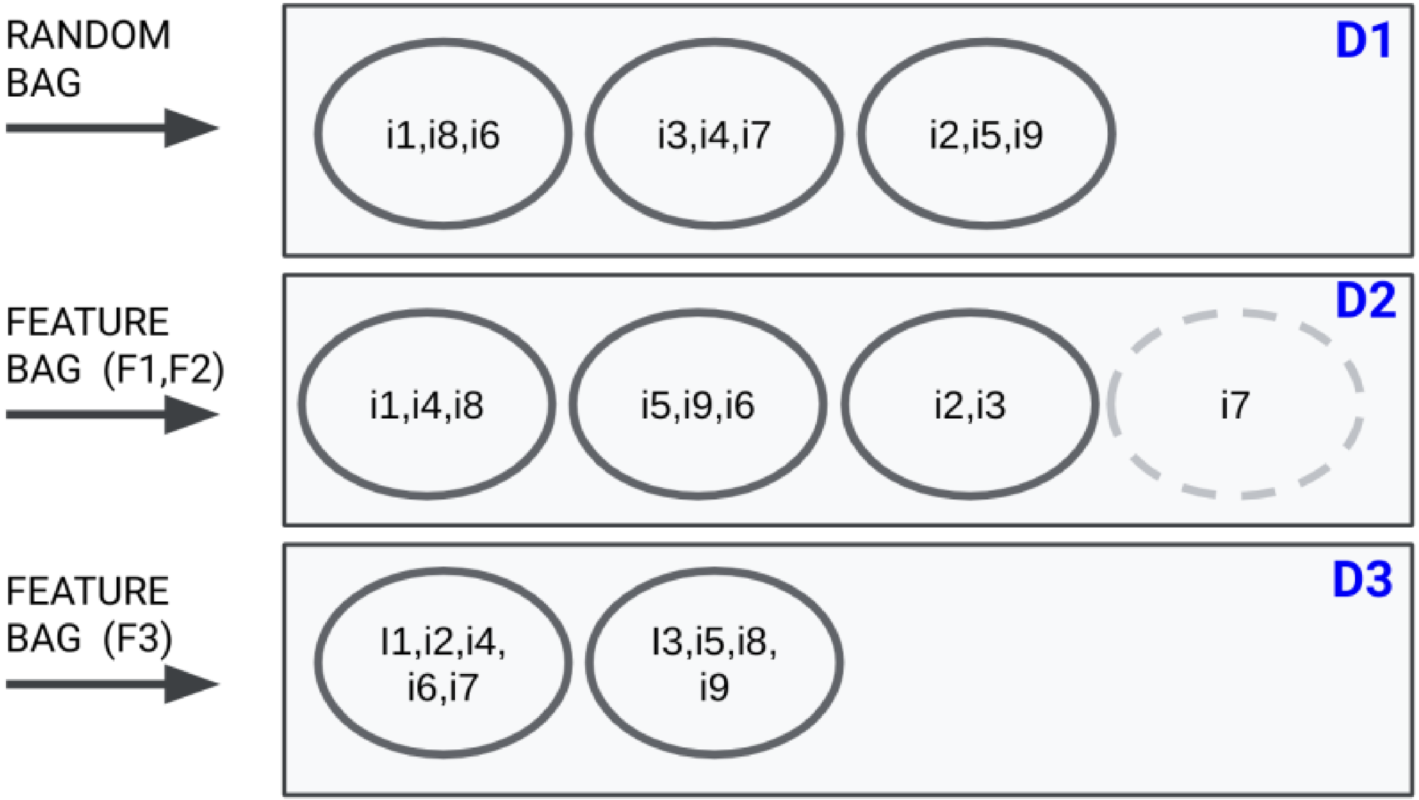
- New learning paradigm
- DP budget scheduling to optimize

ML training from label proportions

Source Instance Dataset

id	F1	F2	F3	Label
i1	8	5	7	1
i2	2	1	7	0
i3	2	1	2	0
i4	8	5	7	1
i5	3	4	2	1
i6	3	4	7	0
i7	4	5	7	0
i8	8	5	2	1
i9	3	4	2	1

Derived Bag Datasets



ML training from label proportions - without DP

- Learning label proportions using proportion matching loss:

$$\min_{h \in \mathcal{H}} \sum_{i=1}^{n} \ell\left(\frac{1}{k}\sum_{j=1}^{k} h(x_{ij}), \alpha_i\right)$$

  - ➢ In general, no guarantee on event-level prediction accuracy

- Learning from label proportions using event-level losses:
  - ➢ Mapping learning from label proportions to learning under label DP + use de-biasing strategy
  - ➢ For binary classification, an unbiased estimator of the event-level loss can be found
  - ➢ If sample size increased by bag size, same performance as event-level learning

With DP : open research question that needs to be investigated

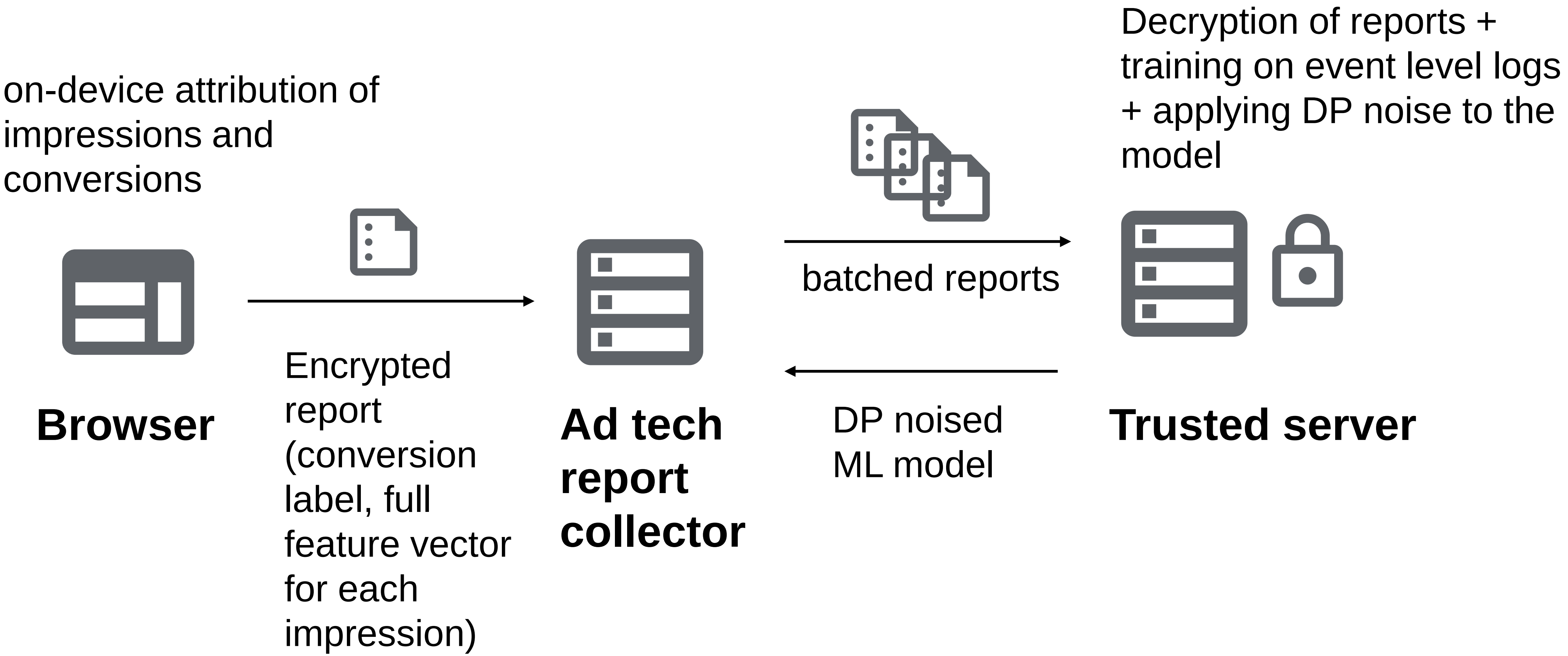
ML training via gradient querying - with DP

- Gradients of generalized linear models (incl. log reg) stand for aggregates
- Adtech companies can query such gradients from aggregation APIs
- Boils down to an online learning problem: main problem being distribution shift
- Global DP amounts to un-biased gradient estimates with larger variance (# SGD)
- Better results than local DP (~ -10% in LLH vs ~ -25% with epsilon = 5) but requires lots of hyperparameter tuning

Agenda

1. Motivation / Context
2. Learning using (sum/count/mean) aggregation APIs
3. Learning using general aggregation APIs
   (# learning in trusted servers)

Understanding of the design



ML training via a trusted server

PROS	CONS
• less noise than local DP • very close to current learning paradigms based conversion labels with event level data	• Likely high trusted server infra cost • to limit noise requires more data, less models, less tasks, ..

Difference with Hybrid proposal

- Encrypted attributed report is **not** a breakdown key combined with the conversion

- Instead this report contains features about the impression, publisher and advertiser user data together with positive or negative label (conversion)

- The aggregated output is not a counter like histogram the full DP noised ML model

What about the used features ?

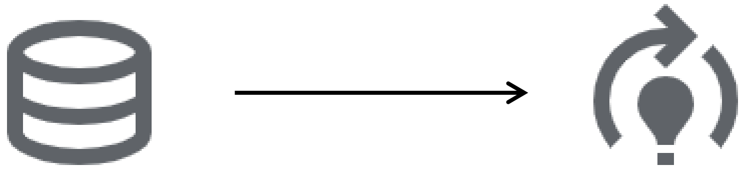
- Sparse feature vector can be contextual information about the bid opportunity (publisher domain, ad slo) and user data (previous impressions, previous advertiser visits)

- One way to inject user data could be with Protected Audience API and modelingSignals field (#1017)

- Even with full contextual campaign optimization results presented here apply

Experimental setup: model & parameters

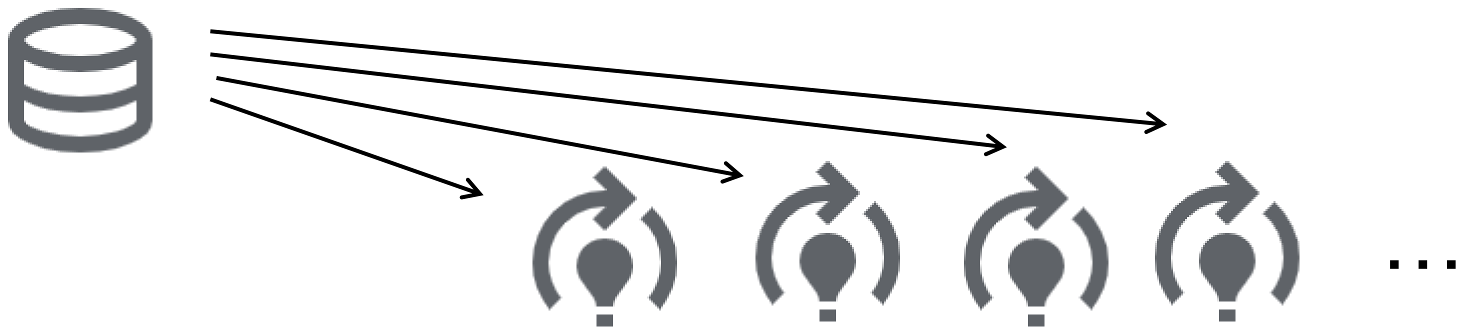
- Logistic regression with SGD optimization
- AdamOptimizer
- Criteo ML challenge dataset with 171 features & 100 mio displays
- 15 epochs =~3 model trainings with model seeding



Number of tasks on same data

~10 different models with different optimization targets
x2 with 1 AB test

For offline tests we may test multiple variations of
those 10 models so up to 50 (rough estimate)



Source code

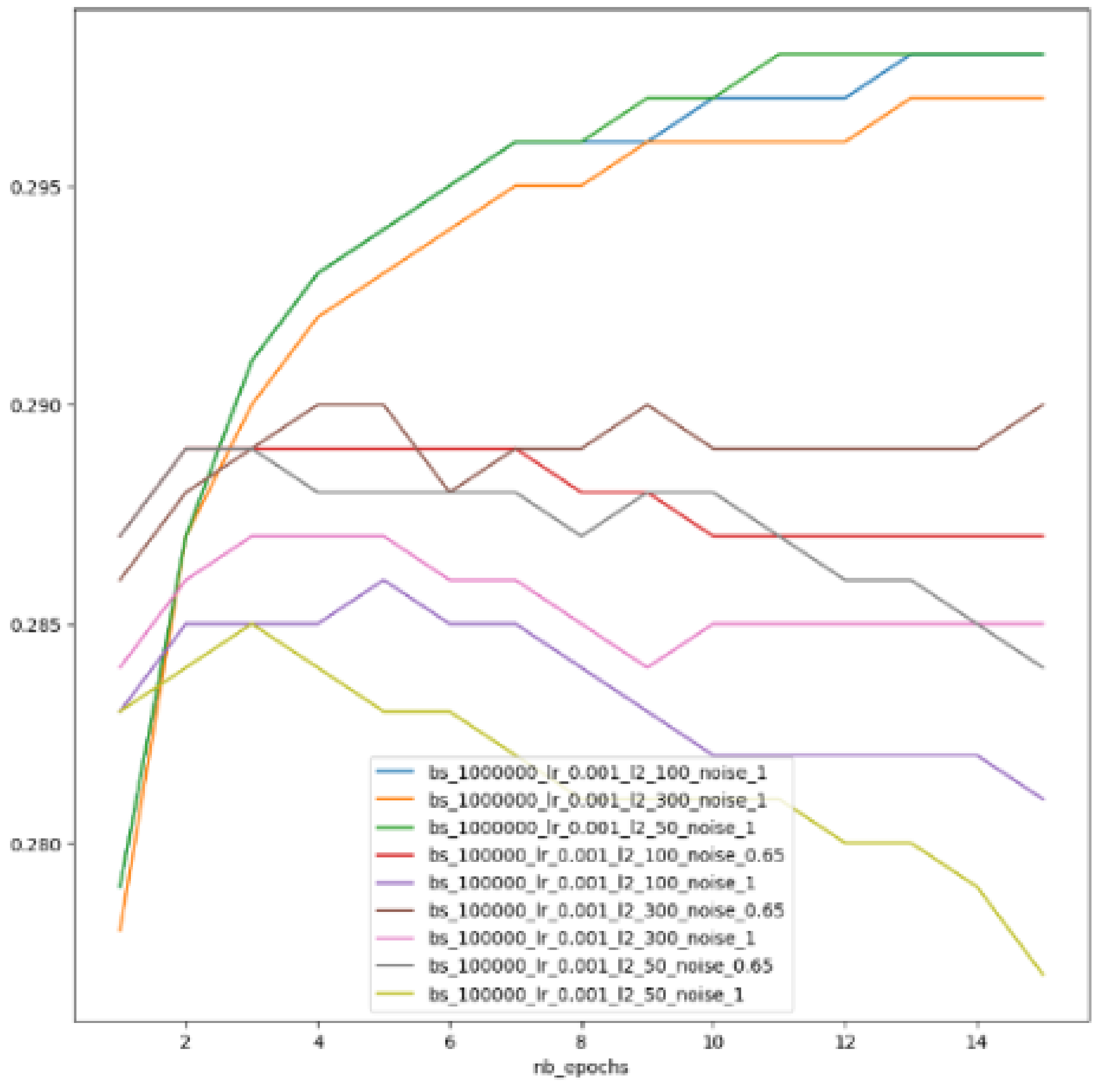
- Noise to the models (gradients) applied via DP-SGD (https://arxiv.org/pdf/1607.00133)
- Epsilon computed with Google DP accountant library here
- Benchmarks executed with different hyper parameters

https://github.com/criteo-research/dp-sgd-ad-click-prediction

How the noise is applied

- Gaussian noise
- Impression level DB budget (similar to ARA)
- Epsilon computed based on Google DP accountant library availabe on github
- Epsilon scales based on
  O(Sqrt(NbModels x NbSteps x NbFeatures)
- Less models, re-learnings or features => less noise
- More data => less noise

Batch size 1mio vs 100k



Target epsilon 5

Large batch sizes bring better performance

Learning of 1 model with best hyper parameters

DP setting	Offline impact (LLHCompVN) 1 model
`epsilon = 1`	-4.0%
`epsilon = 5`	-1.6%
`epsilon = 10`	-1.3%

Learning of 1 model with best hyper parameters

DP setting	Offline impact (LLHCompVN) 1 model
`epsilon = 1`	-4.0%
`epsilon = 5`	-1.6%
`epsilon = 10`	-1.3%

Reasonible for a privacy preserving system

Simulation for learning on multiple tasks (20 models)

- Fixing NbSteps, NbFeatures, NbEvents with -1.6% down lift for 1 model with epsilon 5
- Running 20 learnings on the same data would require epsilon 28 (from 5 => 28)
- Running 20 learnings with 10 times more data would require epsilon 7.24

Conclusion

- It is possible to learn a logistic regression model with good performance and applying global DP constraints

- **28 epsilon seems well above the DP recommendations** for real world applications as the initial assumption on models are not pessimistic at all

- There is a **impact of data size which will favor big players** (epsilon 7 instead of 28 with same downlift for each model)

Next steps

- More experiments needed to reduce epochs for convergence (e.g. increase step size)
- More experiments needed to reduce number of models (e.g. co learning of labels)
- Measure impact of user level DP budgeting as done in “Efficient On-device Budgeting for Differentially-Private Ad-Measurement Systems”
- Public extensive benchmark on production bidding dataset will be released to the community in H1 2025