Privacy Preserving Bandits

Joint work with:

- Mohammad Malekzadeh (QMUL/Brave)
- Hamed Haddadi(ICL/Brave)
- Ben Livshits (ICL/Brave)

Dimitrios Athanasakis (Brave)• 02.03.2020
@dimmu

Why this is an important topic

Personalization is ubiquitous

- Many sites/apps offer personalized experiences
- Advertising (arguably the single biggest application of personalization) fuels the internet.

Personalization is often invasive

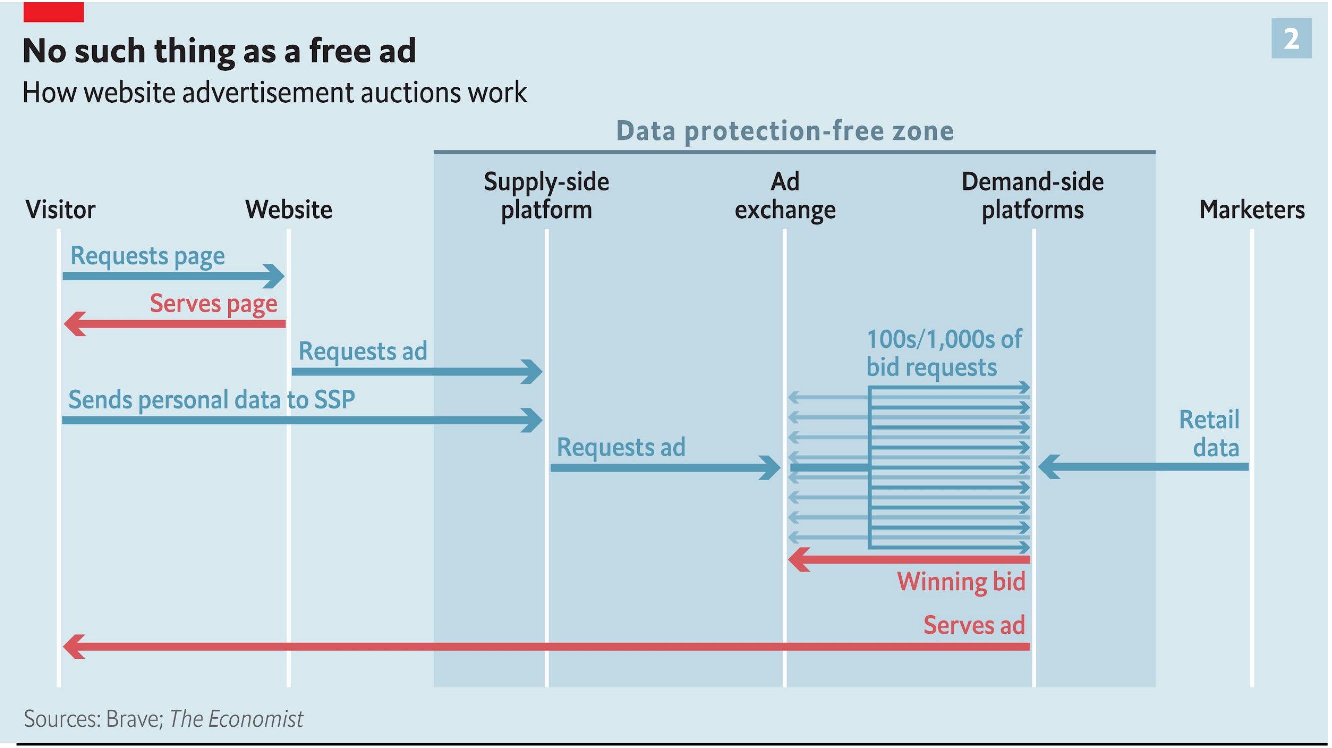
- Tracking all over the internet
- Why is my being a fan of my little pony relevant to the pricing of my plane tickets?
- Some info gets REALLY personal

Real-time Ad bidding

Image source:
The economist

Big tech faces competition and privacy concerns in Brussels

https://www.economist.com/briefing/2019/03/23/big-tech-faces-competition-and-privacy-concerns-in-brussels



Let’s learn everything locally

Great for privacy

- No data ever leaves the user’s device, therefore fewer things to worry from a privacy perspective.
- Eventually the local model will learn a very accurate model recommendation policy for the user.

Not so good for utility

- It may take a long time for the local model to learn a useful recommendation policy
- What happens when new personalization options appear

Online advertising and bandits

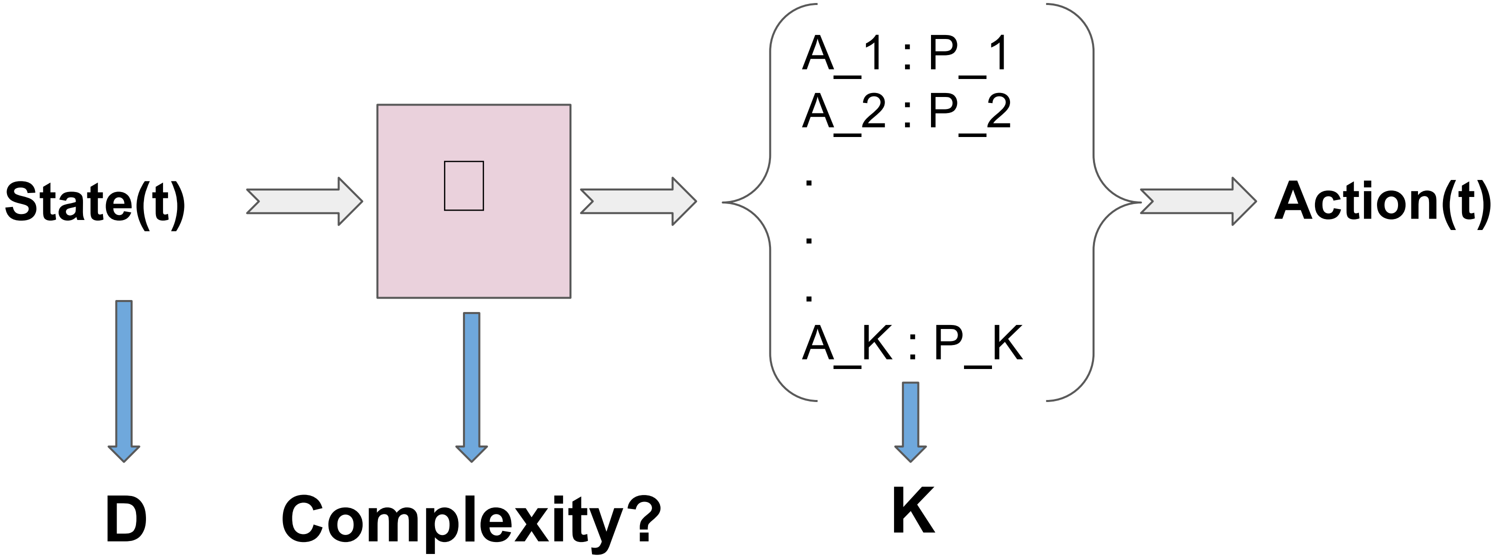
Learning

- What are the user's interests?
- Should we display an ad for product X to user Y?
- Have the interests of the user changed?

Earning

- Given what we know about the user how can we maximise his engagement?

Problem Definition



data tuple = ($S = [S_0, S_1, \ldots, S_D]$, $A \in \{1,2,\ldots,K\}$, $R \in \{0,1\}$)

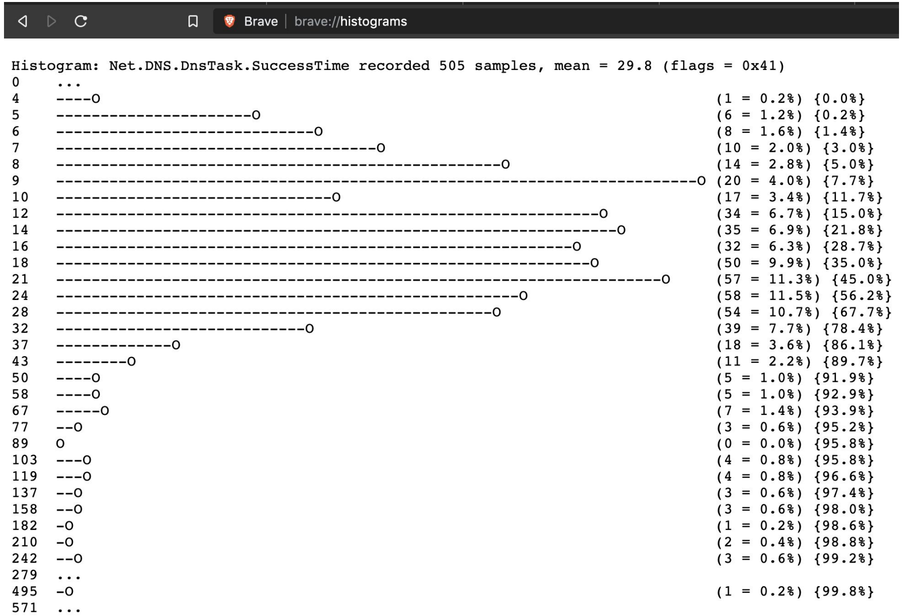
Privacy first!

State? What state?

- **“brave://histograms”**

- **Example:**
    - **Past 100 page visits? (%)**

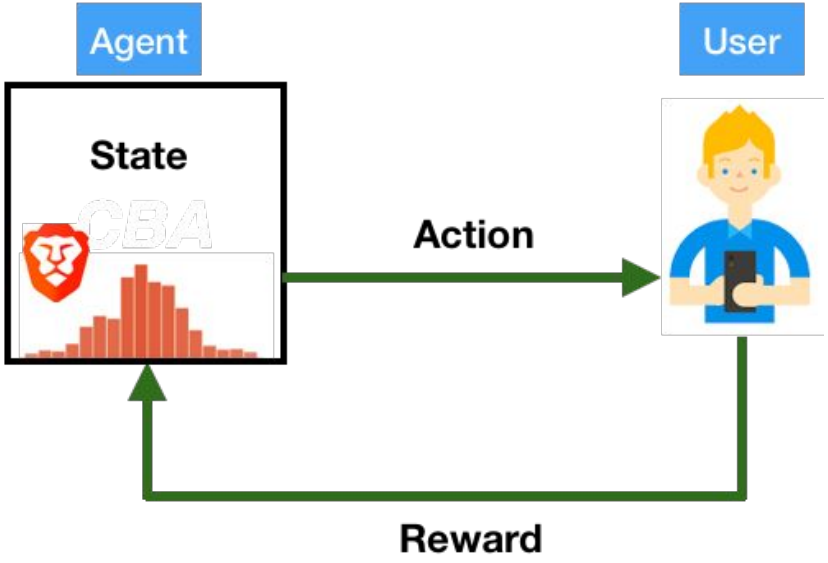
Tech.	Edu.	Fin.	News.	Etc.
0.25	0.15	0.05	0.20	0.35

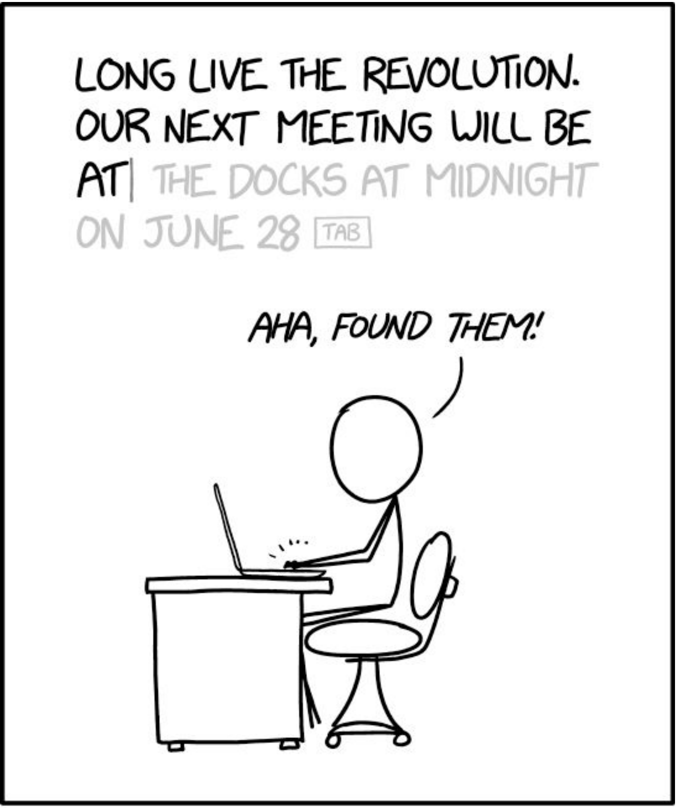



```
Histogram: Net.DNS.DnsTask.SuccessTime recorded 505 samples, mean = 29.8 (flags = 0x41)
0    ...
4    ----O                                                                     (1 = 0.2%) {0.0%}
5    ----------------------O                                                   (6 = 1.2%) {0.2%}
6    -----------------------------O                                            (8 = 1.6%) {1.4%}
7    ------------------------------------O                                     (10 = 2.0%) {3.0%}
8    --------------------------------------------------O                       (14 = 2.8%) {5.0%}
9    ------------------------------------------------------------------------O (20 = 4.0%) {7.7%}
10   -------------------------------O                                          (17 = 3.4%) {11.7%}
12   -------------------------------------------------------------O            (34 = 6.7%) {15.0%}
14   ---------------------------------------------------------------O          (35 = 6.9%) {21.8%}
16   ----------------------------------------------------------O               (32 = 6.3%) {28.7%}
18   ------------------------------------------------------------O             (50 = 9.9%) {35.0%}
21   --------------------------------------------------------------------O     (57 = 11.3%) {45.0%}
24   ----------------------------------------------------O                     (58 = 11.5%) {56.2%}
28   -------------------------------------------------O                        (54 = 10.7%) {67.7%}
32   ----------------------------O                                             (39 = 7.7%) {78.4%}
37   -------------O                                                            (18 = 3.6%) {86.1%}
43   --------O                                                                 (11 = 2.2%) {89.7%}
50   ----O                                                                     (5 = 1.0%) {91.9%}
58   ----O                                                                     (5 = 1.0%) {92.9%}
67   -----O                                                                    (7 = 1.4%) {93.9%}
77   --O                                                                       (3 = 0.6%) {95.2%}
89   O                                                                         (0 = 0.0%) {95.8%}
103  ---O                                                                      (4 = 0.8%) {95.8%}
119  ---O                                                                      (4 = 0.8%) {96.6%}
137  --O                                                                       (3 = 0.6%) {97.4%}
158  --O                                                                       (3 = 0.6%) {98.0%}
182  -O                                                                        (1 = 0.2%) {98.6%}
210  -O                                                                        (2 = 0.4%) {98.8%}
242  --O                                                                       (3 = 0.6%) {99.2%}
279  ...
495  -O                                                                        (1 = 0.2%) {99.8%}
571  ...
```

Research Question

- How we can we enable an agent to know its user **faster** and **better**?
  - Choose the best CBA
  - **Warm start, instead of Cold!**





WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.

Slight Problem

How can we use user data to initialize a warm model without violating a user's privacy?

Can you recognize yourself by your own data?

vs

Vanilla model inversion

vs

Model inversion on noised data

Can we quantify privacy?

Differential Privacy:

Definition 1: Differentially-Private Data Sharing. Given $\epsilon, \delta \geq 0$, we say a data sharing mechanism $\mathcal{M}$ satisfies $(\epsilon, \delta)-$differential privacy if for all pair of neighbor datasets of context vectors $\mathbf{X}, \mathbf{X}'$ differing in only one context vector $\mathbf{x}$ and for all $R \subset Range(\mathcal{M})$,

$$Pr[\mathcal{M}(\mathbf{X}) \in R] \leq e^{\epsilon} Pr[\mathcal{M}(\mathbf{X}') \in R] + \delta$$

(Dwork & Roth 2013)

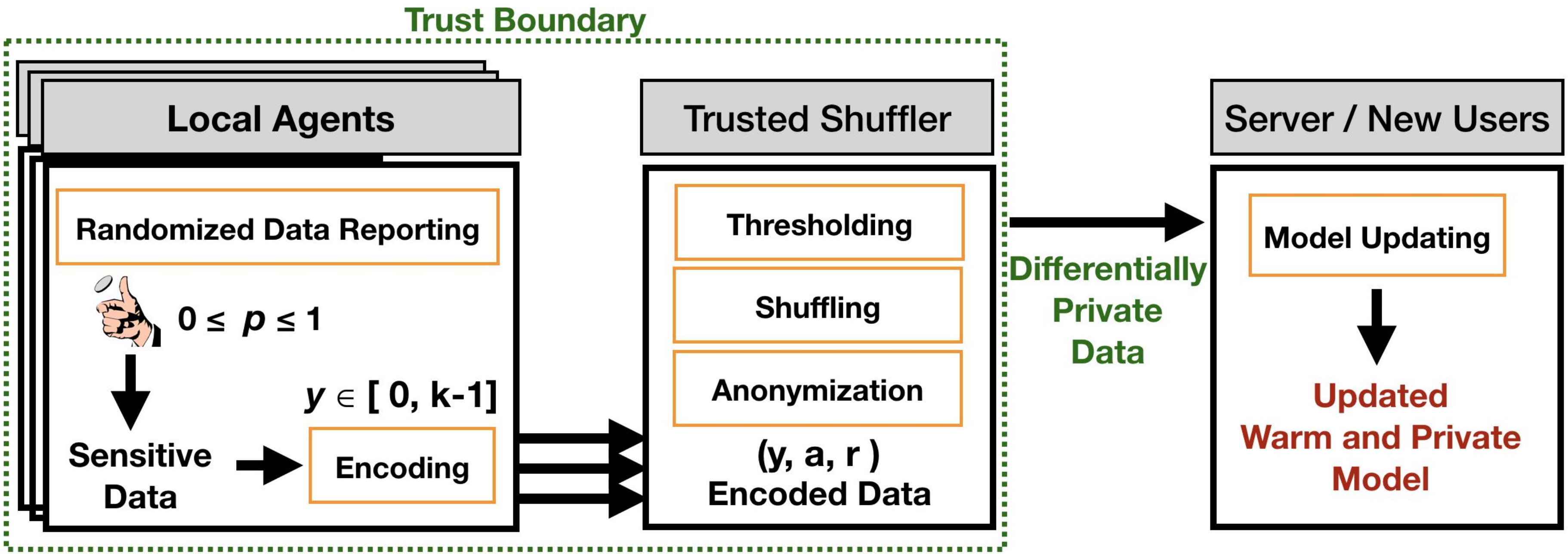
Crowd-blending

Definition 2: Crowd-Blending Encoding. Given $l \geq 1$, we say an encoding mechanism $\mathcal{M}$ satisfies $(l, \bar{\epsilon} = 0)-$crowd-blending privacy if for every context vector $\mathbf{x}$ and for every context dataset $\mathbf{X} = \mathbf{X}' \cup \{\mathbf{x}\}$ we have

$$\left|\{y \in \mathcal{M}(\mathbf{X}) : y = \mathcal{M}(\{\mathbf{x}\})\}\right| \geq l \quad \text{or} \quad \mathcal{M}(\mathbf{X}) = \mathcal{M}(\mathbf{X}')$$

(Gehrke et al 2011)

Our approach: ESA + LinUCB



State Space

- **Histograms**
  - **D**-dimensional vector of real numbers
  - Its sum is **1**
  - It's rounded to **F** decimal points
- e.g. if we set **D=10**:
  - with **F=1** we have **~ 100K** possible states
  - with **F=2** it is **~ 4T**

$$\binom{10^F + D - 1}{D - 1}$$

10^F Stars into D Bars

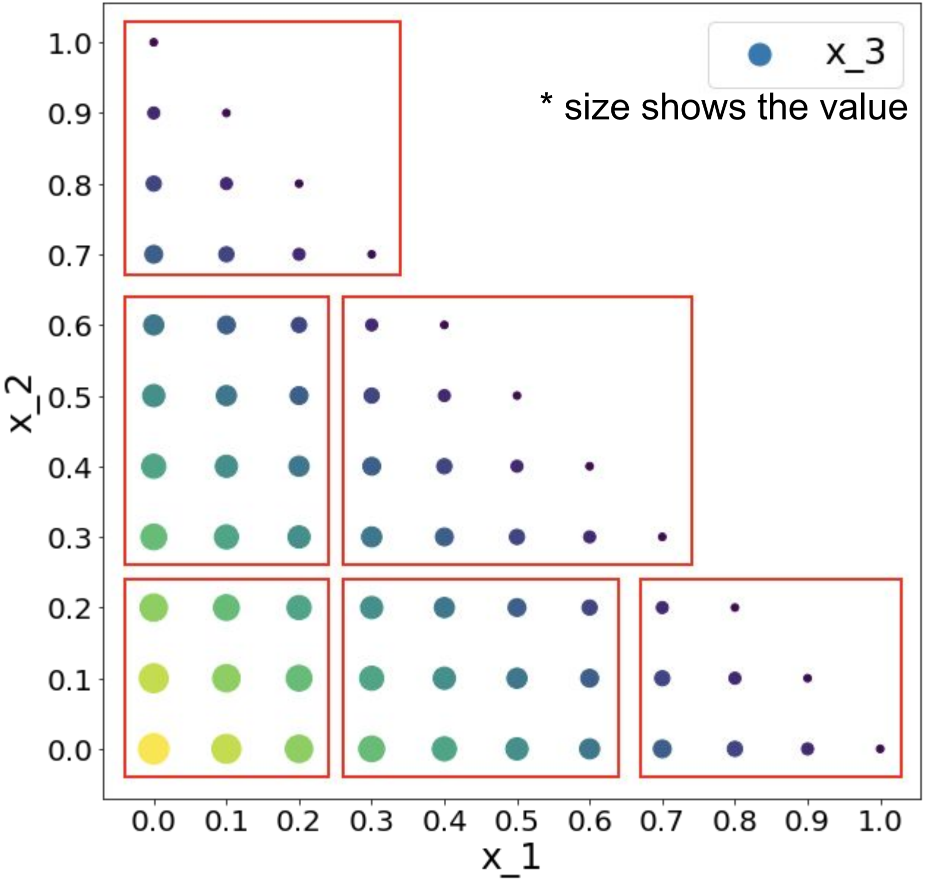
Number of possible states is too large

Encoding

- e.g. **D=3, F=1**
- **66** possible states
- **6 cluster**
  - **Locality-sensitive hashing**
- **3**bits

This helps increasing the size of the crowd a user can blend in.

E.g. D=10 → 10 bits : **4T** → **1K**



Shuffling

- **Anonymization:** Remove Meta-data (eg.ip address) received from local agents
- **Shuffling:** gather tuples received from different sources into batches and shuffle their order.
- **Thresholding:** remove tuples whose encoded context vector frequency in the batch is less than a defined threshold.
- Yes, that means throwing away potentially useful data for the sake of privacy
- This happens in an sgx secure enclave

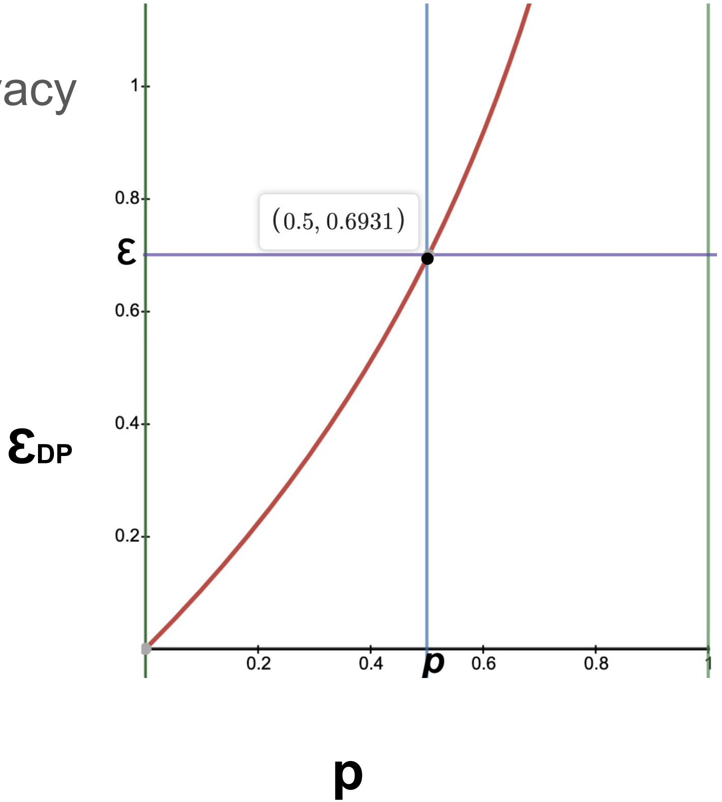
Model updates

- **Updates** are performed using standard LinUCB update rules on the data the shuffler releases.
- Agents can then upload their local models according to the globally updated weights

Privacy Model

- Crowd-Blending + Sampling $\Rightarrow$ Differential Privacy
  - iid random sampling with probability ***p***

$$\varepsilon_{DP} = \ln\left(p \cdot \left(\frac{2-p}{1-p} e^{\varepsilon_{CB}}\right) + (1-p)\right)$$



Evaluation

Algorithm

- **Linear UCB**

Context

- **Histograms**

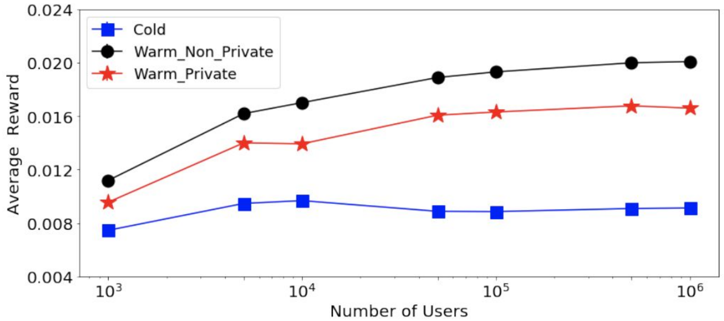
Environment

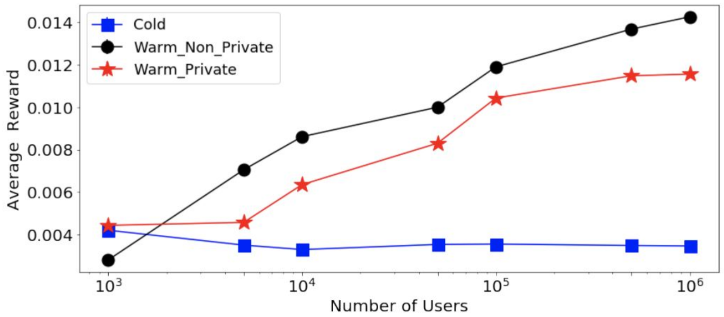
- **Synthetic Datasets**
  - Linear and nonlinear randomly initialized mapping functions
    - Input: a histogram
    - Output: a stochastic preference model
- **Real Multi-Label Datasets**
  - Input: a binary vector (features)
  - Output: a binary vector (labels)
- **Criteo Ad Recommendation Dataset**
  - Input: Integer values (unknown features)
  - Output: a one-hot vector (product category)

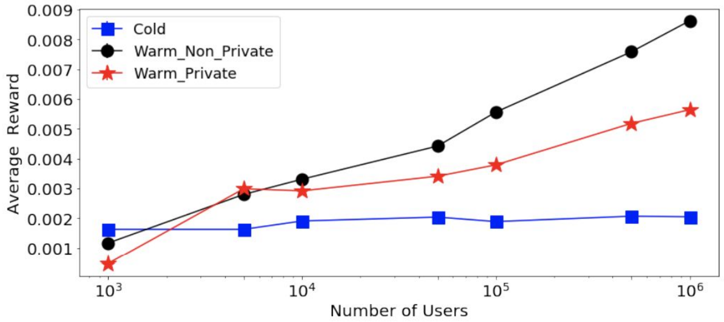
Github:
https://github.com/mmalekzadeh/privacy-preserving-bandits

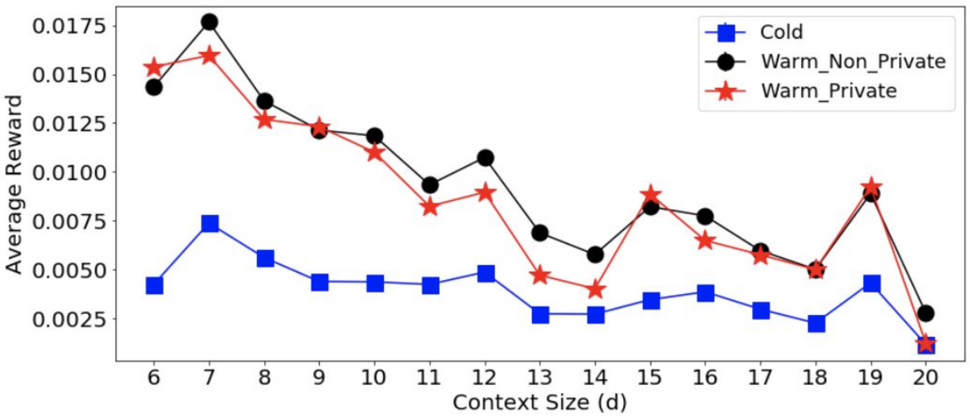
Results: Synthetic Data

- Left: effect of available actions on expected reward for varying numbers of users
- Bottom: effect of the dimensionality of the context on expected reward

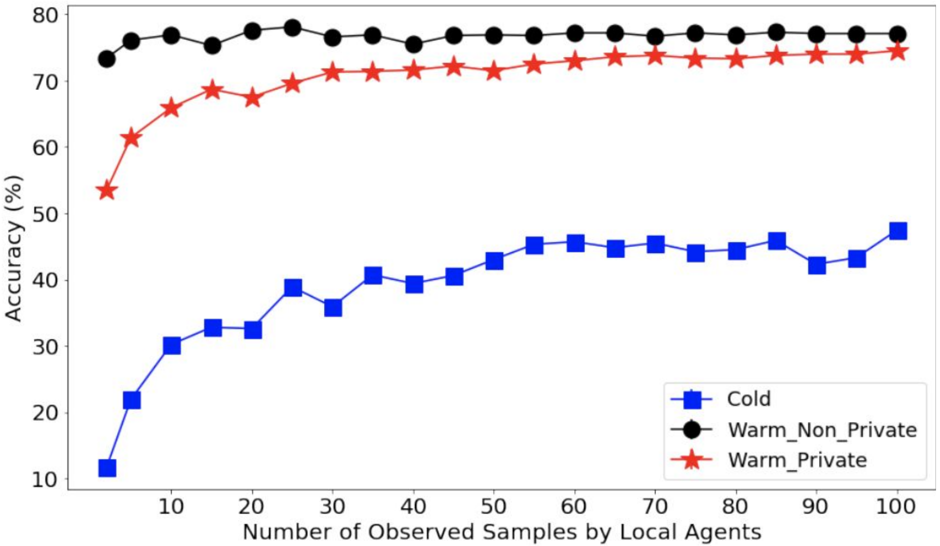


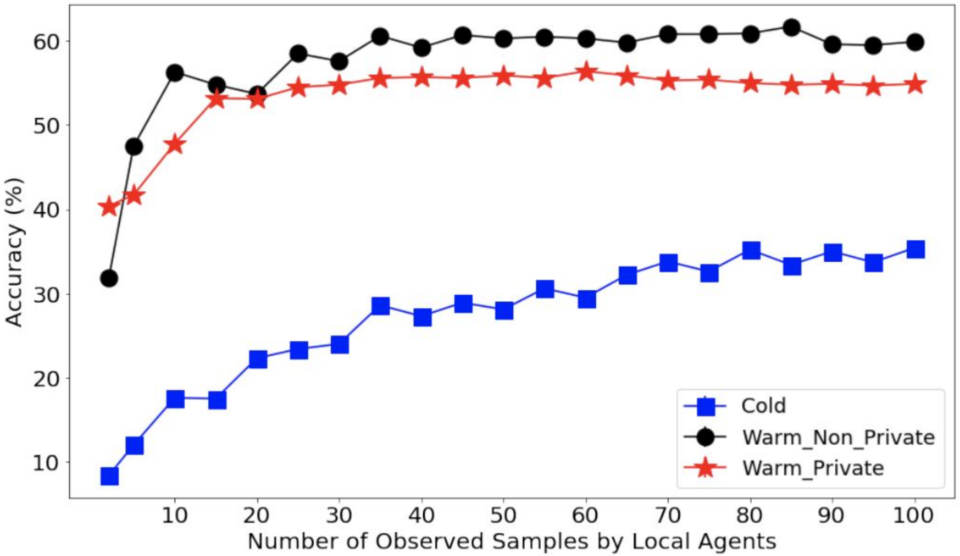






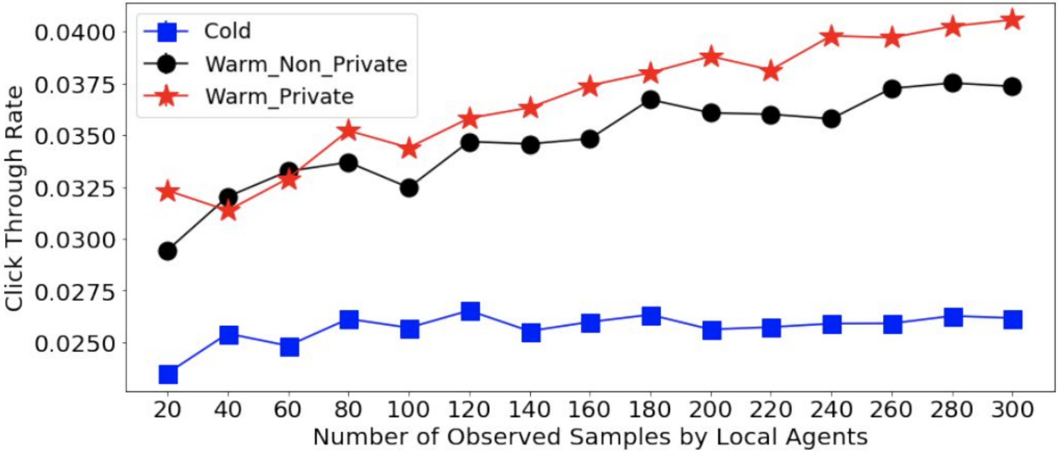
Results: Multi-Label Classification

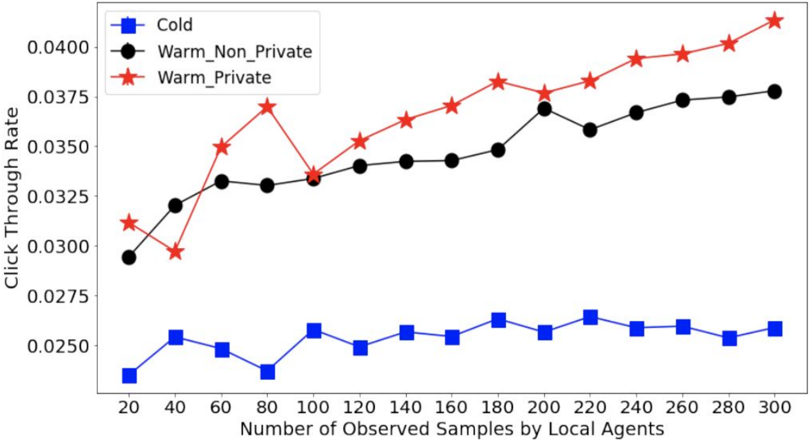




- MediaMill: d=20, |A|=40, ~ 44000 instances
- TextMining: d=20, |A|=20, ~28,500 instances

Results: Ad. Recommendation (Criteo)





- k= 32
- k= 128

|A|=40, d=10, u=3,000 agents

Some Remarks

- The Criteo ad recommendation experiments are somewhat strange but surely interesting
- ESA is making a comeback (ESA Revisited)
- Also SMPC for bandits
- Feel free to play around with the notebooks. Also stickers, again

Github:
https://github.com/mmalekzadeh/privacy-preserving-bandits

Personal Notes

- Mohammad will be looking for a job soon.
- Pleasantly surprised to see some remote presentations.

Let's keep in touch

1. Poster #15
2. Working on privacy? Let's talk. Have experiences in the adtech ecosystem? We'd like to hear from you.
3. We're always looking for great engineers: https://brave.com/careers/

Also @dimmu