Automatic, Fine-Grained Algorithmic Choice for Differential Privacy

Jacob Imola

Carnegie Mellon University

2017

Motivation

Netflix wants to anonymize database and release it:

	Titanic	Date	The Notebook	Date
Jordan	8.5	1/17	7.5	3/20
Jean	6	3/5	5	3/7
Scott	2	1/21	9	2/26
Serena	4.5	4/5	9	3/7

Motivation

Just cross out names!

	Titanic	Date	The Notebook	Date
User 1	8.5	1/17	7.5	3/20
User 2	6	3/5	5	3/7
User 3	2	1/21	9	2/26
User 4	4.5	4/5	9	3/7

Motivation

Just cross out names!

	Titanic	Date	The Notebook	Date
User 1	8.5	1/17	7.5	3/20
User 2	6	3/5	5	3/7
User 3	2	1/21	9	2/26
User 4	4.5	4/5	9	3/7

Months later...

Motivation

Netflix

	Titanic	Date	The Notebook	Date
User 1	8.5	1/17	7.5	3/20
User 2	6	3/5	5	3/7
User 3	2	1/21	9	2/26
User 4	4.5	4/5	9	3/7

IMDB

Scott on 1/22:
(1.5/5) Titanic was terrible!

Jordan on 1/20:
(4.0/5) Enjoyed Titanic!

Jean on 3/6:
(2.0/5) The Notebook was pretty overrated :/

Motivation

Netflix

	Titanic	Date	The Notebook	Date
User 1	8.5	1/17	7.5	3/20
User 2	6	3/5	5	3/7
User 3	2	1/21	9	2/26
User 4	4.5	4/5	9	3/7

IMDB

Scott on 1/22:
(1.5/5) Titanic was terrible!

Jordan on 1/20:
(4.0/5) Enjoyed Titanic!

Jean on 3/6:
(2.0/5) The Notebook was pretty overrated :/

Netflix promised that all movie ratings would be protected!

Differential Privacy

Definition

For all D and D' differing in 1 row: P is ϵ-DP if $\Pr(P(D) = O) < e^{\epsilon} \Pr(P(D') = O)$ for all O.

$$P\underbrace{\begin{pmatrix} \text{Jordan} & 8.5 & 1/17 & 7.5 & 3/20 \\ \text{Jean} & 6 & 3/5 & 5 & 3/7 \\ \text{Scott} & 2 & 1/21 & 9 & 2/26 \\ \text{Serena} & 4.5 & 4/5 & 9 & 3/7 \end{pmatrix}}_{D} = \underbrace{\begin{matrix} \text{User 1} & 8.5 & 1/17 & 7.5 & 3/20 \\ \text{User 2} & 6 & 3/5 & 5 & 3/7 \\ \text{User 3} & 2 & 1/21 & 9 & 2/26 \\ \text{User 4} & 4.5 & 4/5 & 9 & 3/7 \end{matrix}}_{O}$$

$$P\underbrace{\begin{pmatrix} \text{Jordan} & 8.5 & 1/17 & 7.5 & 3/20 \\ \text{Jean} & 6 & 3/5 & 5 & 3/7 \\ \text{Scott} & 2 & 1/21 & 9 & 2/26 \\ \text{Serena} & 8.5 & 4/6 & 3 & 1/20 \end{pmatrix}}_{D'} = \underbrace{\begin{matrix} \text{User 1} & 8.5 & 1/17 & 7.5 & 3/20 \\ \text{User 2} & 6 & 3/5 & 5 & 3/7 \\ \text{User 3} & 2 & 1/21 & 9 & 2/26 \\ \text{User 4} & 8.5 & 4/6 & 3 & 1/20 \end{matrix}}_{O'}$$

Differential Privacy

Definition

For all D and D' differing in 1 row: P is ϵ-DP if $\Pr(P(D) = O) < e^{\epsilon} \Pr(P(D') = O)$ for all O.

$$P\underbrace{\begin{pmatrix} \text{Jordan} & 8.5 & 1/17 & 7.5 & 3/20 \\ \text{Jean} & 6 & 3/5 & 5 & 3/7 \\ \text{Scott} & 2 & 1/21 & 9 & 2/26 \\ \text{Serena} & 4.5 & 4/5 & 9 & 3/7 \end{pmatrix}}_{D} = \underbrace{\begin{matrix} \text{User 1} & 8.5 & 1/17 & 7.5 & 3/20 \\ \text{User 2} & 6 & 3/5 & 5 & 3/7 \\ \text{User 3} & 2 & 1/21 & 9 & 2/26 \\ \text{User 4} & 4.5 & 4/5 & 9 & 3/7 \end{matrix}}_{O}$$

$$P\underbrace{\begin{pmatrix} \text{Jordan} & 8.5 & 1/17 & 7.5 & 3/20 \\ \text{Jean} & 6 & 3/5 & 5 & 3/7 \\ \text{Scott} & 2 & 1/21 & 9 & 2/26 \\ \text{Serena} & 8.5 & 4/6 & 3 & 1/20 \end{pmatrix}}_{D'} = \underbrace{\begin{matrix} \text{User 1} & 8.5 & 1/17 & 7.5 & 3/20 \\ \text{User 2} & 6 & 3/5 & 5 & 3/7 \\ \text{User 3} & 2 & 1/21 & 9 & 2/26 \\ \text{User 4} & 8.5 & 4/6 & 3 & 1/20 \end{matrix}}_{O'}$$

Violation: $\Pr(P(D) = O) = 1$ and $\Pr(P(D') = O) = 0$

Example

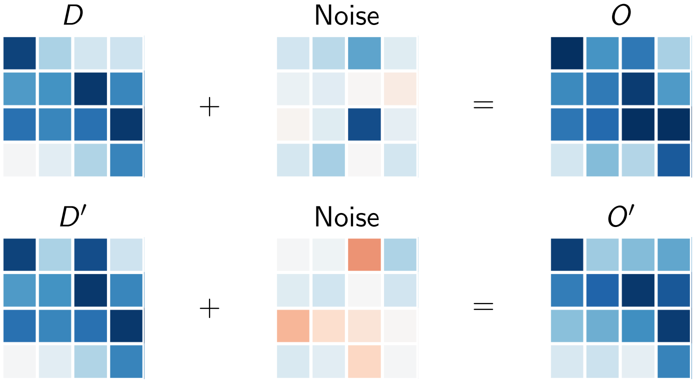
A representation change

User 1	8.5	1/17	7.5	3/20
User 2	6	3/5	5	3/7
User 3	2	1/21	9	2/26
User 4	4.5	4/5	9	3/7

$\rightarrow$

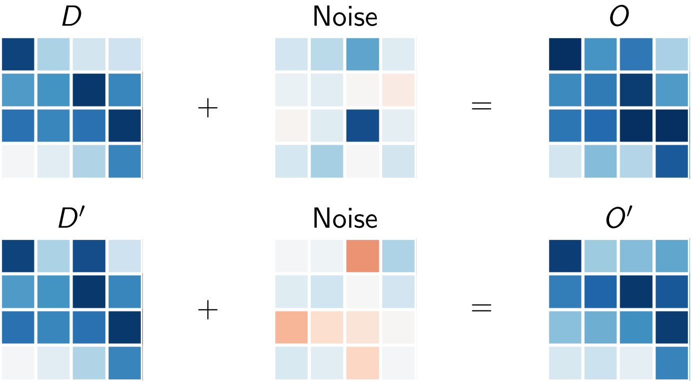
Example

Method 1



Example

Method 1



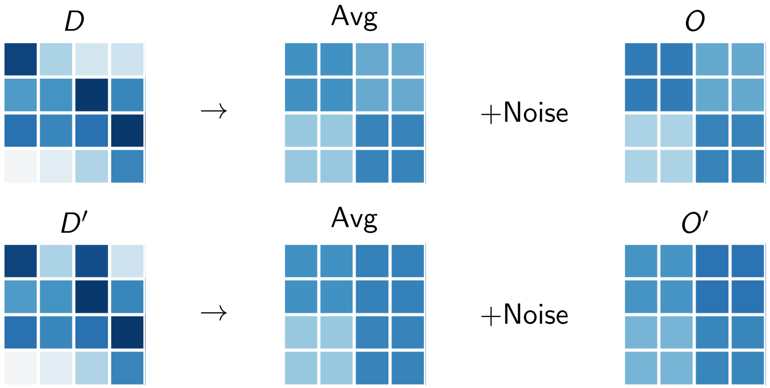
$\Pr(P(D) = O) \approx 10^{-8}$ $\qquad$ $\Pr(P(D') = O) \approx 2 \times 10^{-9}$

Seeing O, attacker cannot distinguish D and D'.

Example

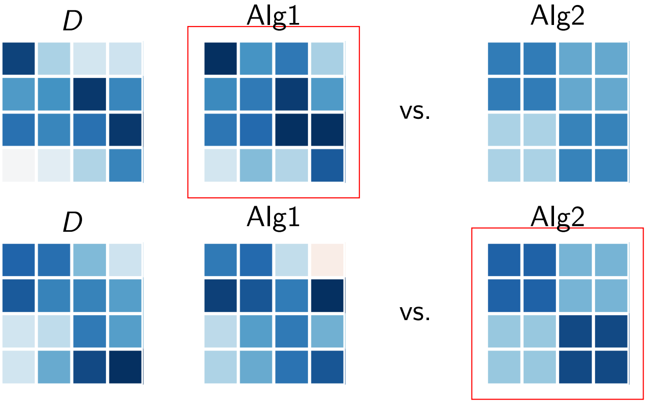
Method 2

- Sum into 4 buckets, add noise, divide by 4



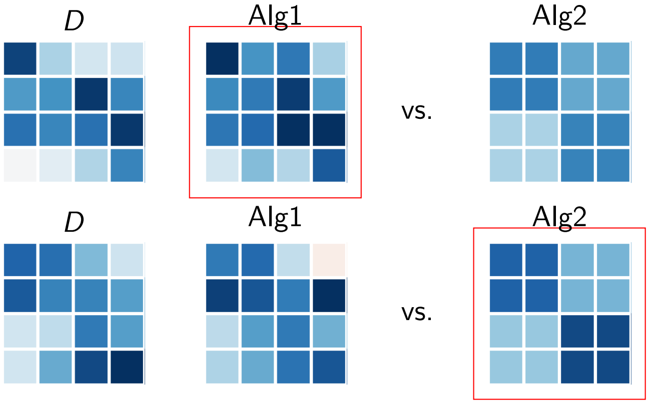
Example

Which is better?



Example

Which is better?



DP complicates algorithm analysis due to noise, makes algorithm deployment hard.

Vision

Task: Remove burden of DP algorithm analysis.

1. **Correctness** Differential privacy is never violated.
2. **Generalizability** Works on arbitrary code.
3. **Performance** Makes choice “close enough” to optimal.

Vision

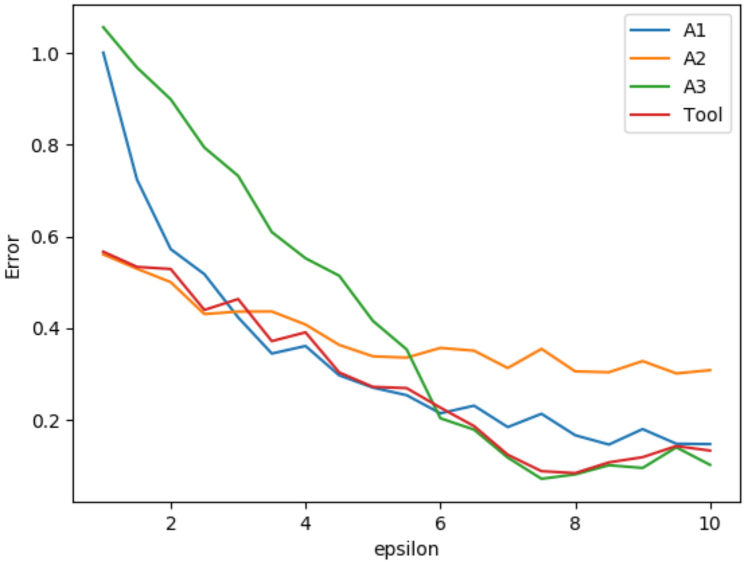
Task: Remove burden of DP algorithm analysis.

1. **Correctness** Differential privacy is never violated.
2. **Generalizability** Works on arbitrary code.
3. **Performance** Makes choice “close enough” to optimal.

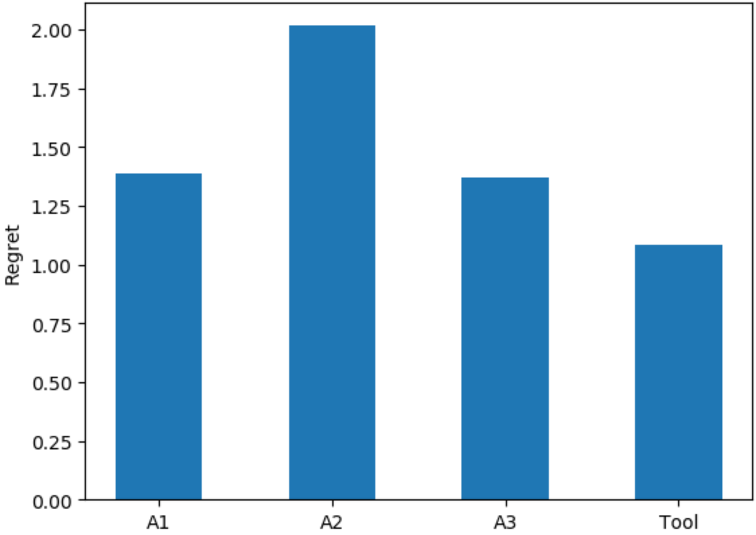
Solution: A programming language!

```
answerHistQueries = MkChoiceMaker among {Alg1, Alg2}
answers = answerHistQueries(data, queries)
```

Note on performance



Note on performance

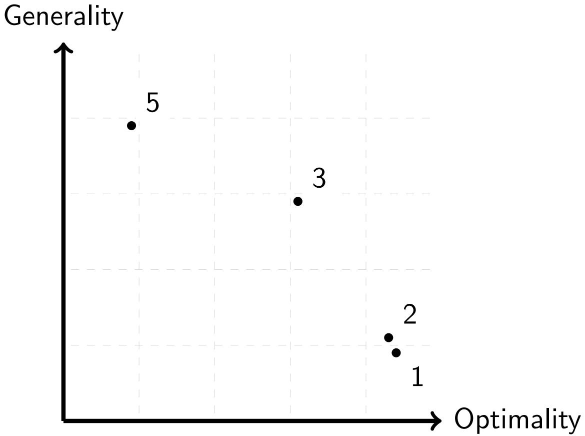


Challenges

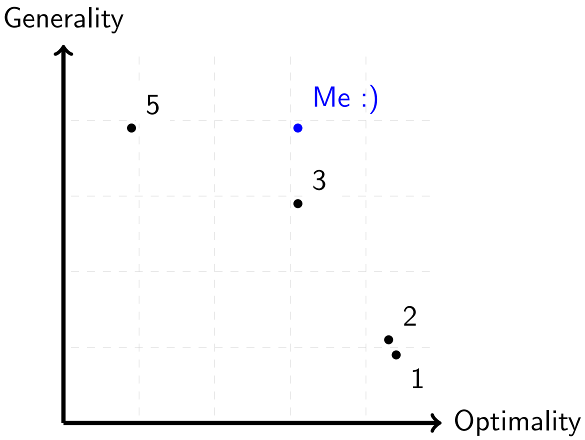
```
answerHistQueries = MkChoiceMaker among {Alg1, Alg2}
answers = answerHistQueries(data, queries)
```

- Generality $\implies$ Can only run Alg1, Alg2
- Meta-machine learning: function $f : DB \to Alg$
- Intractable—data science cannot be automated well.

Existing Work

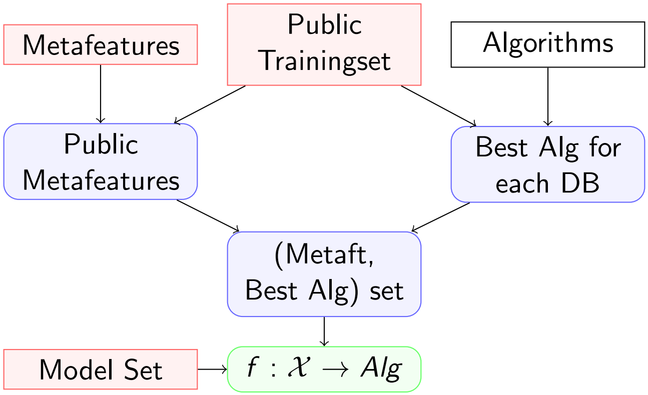


Existing Work



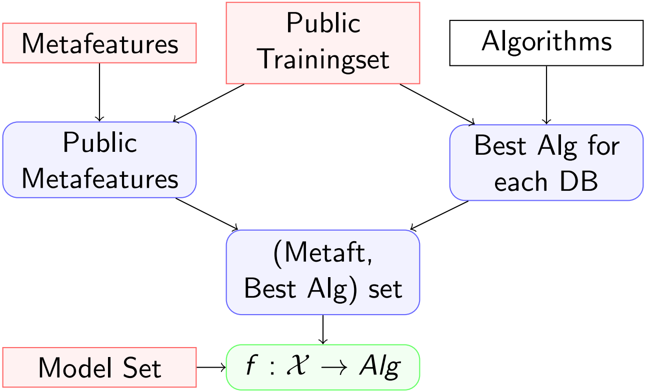
Solution Overview

- Metafeatures modeled after data science approach
- $f : DB \rightarrow Alg$ becomes $f : \mathcal{X} \rightarrow Alg$.



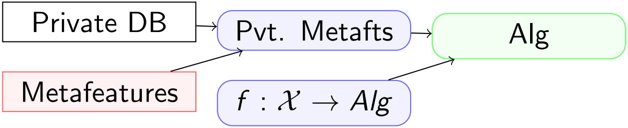
Solution Overview

- Metafeatures modeled after data science approach
- $f : DB \rightarrow Alg$ becomes $f : \mathcal{X} \rightarrow Alg$.

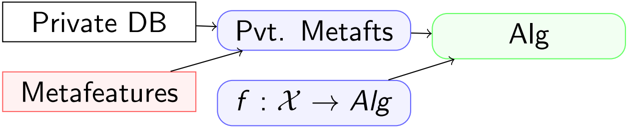


Important: Trainingset must have lots of DB's for training!

Solution Overview



Solution Overview



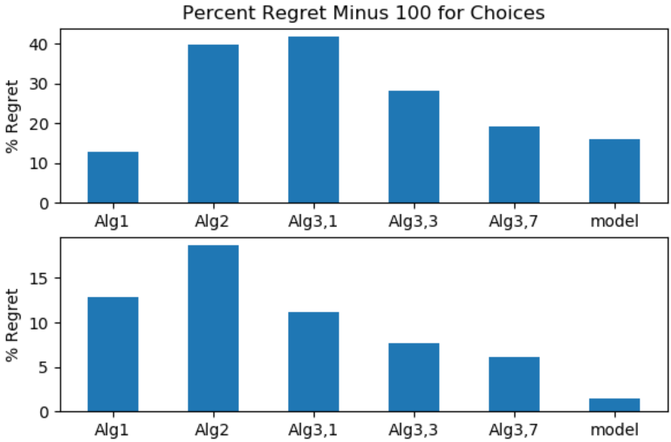

```
answerHistQueries =
MkChoiceMaker among {Alg1, Alg2}
 informed by {dbSize, dbNumRows}
 modeled by LinearModel with ErrorFunc
 trained on TrainingSet}

answers = answerHistQueries(data, queries)
```

Experimental Setup

- **Algorithms** Stopping Criteria for Private Decision Trees.
- **Metafeatures** DB size, epsilon, domain size.
- **Classification** Linear Classifiers
- **Training Set**
  1. 300 real DB snapshots, 100 real DB snapshots
  2. 300 synth. DB snapshots, 100 synth. DB snapshots

Results


Percent Regret Minus 100 for Choices
% Regret
40
30
20
10
0
Alg1
Alg2
Alg3,1
Alg3,3
Alg3,7
model
% Regret
15
10
5
0
Alg1
Alg2
Alg3,1
Alg3,3
Alg3,7
model

Conclusion

- Automated meta-ML training and algorithm deployment
- Automating data science is hard
- Future work: Automate a more sophisticated data science workflow