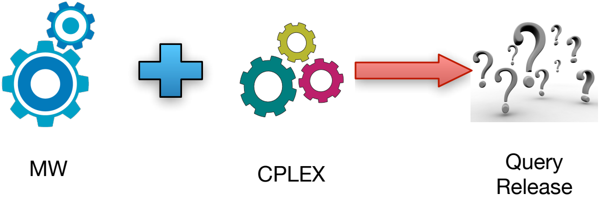
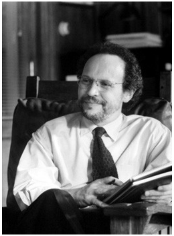
Dual Query Release

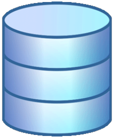


Marco Gaboardi, Emilio Gallego Jésus Arias,
Justin Hsu, Aaron Roth, Steven Wu

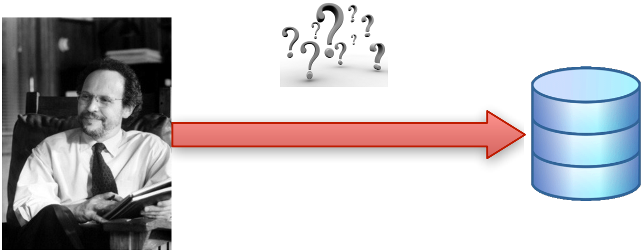
University of Pennsylvania

December 11th, 2013

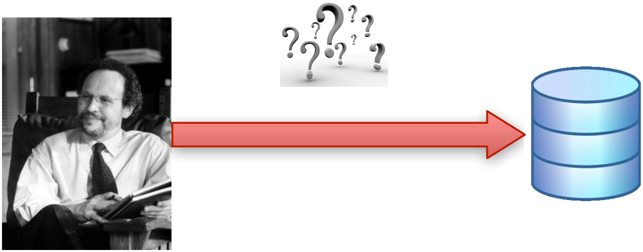




General setting

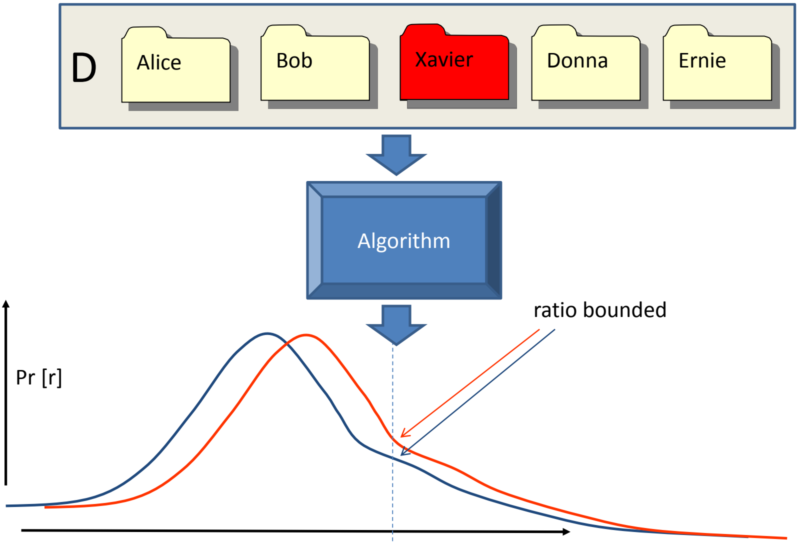


General setting



Analyst wants answers to a bunch of queries, preserving privacy.

Differential privacy [DMNS]



Definition (DMNS)

Let M be a randomized mechanism from databases to range $\mathcal{R}$, and let D, D' be databases differing in one record. M is (ε, δ)-differentially private if for every $r \in \mathcal{R}$,

$$\Pr[M(D) = r] \leq e^{\varepsilon} \cdot \Pr[M(D') = r] + \delta.$$

Useful properties

- Very strong, worst-case privacy guarantee
- Well-behaved under composition, post-processing

The problem

Query release

- Space of possible records $\mathcal{X} = \{0, 1\}^d$ (d binary attributes)
- Database $D \in \mathbb{N}^{|\mathcal{X}|}$ of n records (histogram)
- Analysts want accurate answers to a large (exponential in n) set $\mathcal{Q}$ of counting queries

Query release

- Space of possible records $\mathcal{X} = \{0,1\}^d$ (d binary attributes)
- Database $D \in \mathbb{N}^{|\mathcal{X}|}$ of n records (histogram)
- Analysts want accurate answers to a large (exponential in n) set $\mathcal{Q}$ of counting queries

"What fraction of records satisfy P?"

The problem

Query release

- Space of possible records $\mathcal{X} = \{0,1\}^d$ (d binary attributes)
- Database $D \in \mathbb{N}^{|\mathcal{X}|}$ of n records (histogram)
- Analysts want accurate answers to a large (exponential in n) set $\mathcal{Q}$ of counting queries

"What fraction of records satisfy P?"

- Goal: privately construct distribution $\hat{D}$ approximating D

Approaches from learning theory

- Dwork, Rothblum, Vadhan: query release via boosting
- Hardt and Rothblum: MW algorithm for query release
- Experimentally evaluated by Hardt, Ligett, McSherry
- Performs well for $\lesssim 80$ binary attributes

High-dimensional data

Approaches from learning theory

- Dwork, Rothblum, Vadhan: query release via boosting
- Hardt and Rothblum: MW algorithm for query release
- Experimentally evaluated by Hardt, Ligett, McSherry
- Performs well for $\lesssim 80$ binary attributes

What is the bottleneck?

- Operate on distribution over all possible records
- For $d \geq 100$, more than $2^{100} \sim 10^{30}$ records

Is it possible to do better?

Is it possible to do better?

In general, no.

- Impossibility results (see [DNRRV], [Ullman-Vadhan], or [Ullman])
- Exponentially large collection of queries can't be answered efficiently and accurately

Is it possible to do better?

In general, no.

- Impossibility results (see [DNRRV], [Ullman-Vadhan], or [Ullman])
- Exponentially large collection of queries can't be answered efficiently and accurately

Our approach

- Reconfigure existing algorithms to isolate hard step
- Theoretically hard, but often tractable in practice

Today

1. Query release as a zero sum game
2. Finding equilibrium of this game
3. Dual query release algorithm
4. Performance

The players

- Data player: actions are records in $\mathcal{X}$
- Query player: actions are queries in $\mathcal{Q}$

The players

- Data player: actions are records in $\mathcal{X}$
- Query player: actions are queries in $\mathcal{Q}$

The payoffs/losses

- If data plays $r \in \mathcal{X}$ and query plays $q \in \mathcal{Q}$, payoff

$$q(r) - q(D)$$

- Data player minimizes, query player maximizes (zero sum)

"Database with one record"

The players

- Data player: actions are <u>records in $\mathcal{X}$</u>
- Query player: actions are <u>queries in $\mathcal{Q}$</u>

The payoffs/losses

- If data plays $r \in \mathcal{X}$ and query plays $q \in \mathcal{Q}$, payoff

$$q(r) - q(D)$$

- Data player minimizes, query player maximizes (zero sum)

The query release game

The players

“Database with one record”

- Data player: actions are records in $\mathcal{X}$
- Query player: actions are queries in $\mathcal{Q}$

“Query with high error”

The payoffs/losses

- If data plays $r \in \mathcal{X}$ and query plays $q \in \mathcal{Q}$, payoff

$$q(r) - q(D)$$

- Data player minimizes, query player maximizes (zero sum)

The players

“Database with one record”

- Data player: actions are records in $\mathcal{X}$
- Query player: actions are queries in $\mathcal{Q}$

“Query with high error”

The payoffs/losses

- If data plays $r \in \mathcal{X}$ and query plays $q \in \mathcal{Q}$, payoff

$$q(r) - q(D)$$

“Error” (D is true database)

- Data player minimizes, query player maximizes (zero sum)

Approximate equilibrium

Definition

- Distributions $\hat{D}$ over records, $\hat{Q}$ over queries
- Players gain at most α by playing another distribution

Definition

- Distributions $\hat{D}$ over records, $\hat{Q}$ over queries
- Players gain at most α by playing another distribution

For query release

- If data player plays D (true database), expects zero payoff versus any query
- At α-approximate equilibrium, $\hat{D}$ has expected error at most α on any query in $\mathcal{Q}$

Approximate equilibrium

Definition

- Distributions $\hat{D}$ over records, $\hat{Q}$ over queries
- Players gain at most α by playing another distribution

For query release

- If data player plays D (true database), expects zero payoff versus any query
- At α-approximate equilibrium, $\hat{D}$ has expected error at most α on any query in $\mathcal{Q}$

Synthetic data
for query release

Primal approach

- Manipulate candidate database (distribution over records $\mathcal{X}$)
- Optimize: find query in $\mathcal{Q}$ with high error
- Well-studied idea [HR, HLM, ...]

Computing the equilibrium

Primal approach

- Manipulate candidate database (distribution over records $\mathcal{X}$)
- Optimize: find query in $\mathcal{Q}$ with high error
- Well-studied idea [HR, HLM, ...]

Dual approach

- Manipulate distribution over queries $\mathcal{Q}$
- Optimize: find record in $\mathcal{X}$ with low error
- Very similar to Dwork, Rothblum, Vadhan

Computing the equilibrium

Primal approach

Big, intractable

- Manipulate candidate database (distribution over records $\mathcal{X}$)
- Optimize: find query in $\mathcal{Q}$ with high error
- Well-studied idea [HR, HLM, ...]

Dual approach

- Manipulate distribution over queries $\mathcal{Q}$
- Optimize: find record in $\mathcal{X}$ with low error
- Very similar to Dwork, Rothblum, Vadhan

Computing the equilibrium

Primal approach

Big, intractable

- Manipulate candidate database (distribution over records $\mathcal{X}$)
- Optimize: find query in $\mathcal{Q}$ with high error

Small, tractable

- Well-studied idea [HR, HLM, ...]

Dual approach

- Manipulate distribution over queries $\mathcal{Q}$
- Optimize: find record in $\mathcal{X}$ with low error

- Very similar to Dwork, Rothblum, Vadhan

Primal approach

Big, intractable

- Manipulate candidate database (distribution over records $\mathcal{X}$)
- Optimize: find query in $\mathcal{Q}$ with high error

Small, tractable

- Well-studied idea [HR, HLM, ...]

Dual approach

Small, tractable

- Manipulate distribution over queries $\mathcal{Q}$
- Optimize: find record in $\mathcal{X}$ with low error

- Very similar to Dwork, Rothblum, Vadhan

Computing the equilibrium

Primal approach

Big, intractable

- Manipulate candidate database (distribution over records $\mathcal{X}$)
- Optimize: find query in $\mathcal{Q}$ with high error

Small, tractable

- Well-studied idea [HR, HLM, ...]

Dual approach

Small, tractable

- Manipulate distribution over queries $\mathcal{Q}$
- Optimize: find record in $\mathcal{X}$ with low error

Big, intractable

- Very similar to Dwork, Rothblum, Vadhan

Computing the equilibrium

Primal approach

Big, intractable

- Manipulate candidate database (distribution over records $\mathcal{X}$)
- Optimize: find query in $\mathcal{Q}$ with high error

Small, tractable

- Well-studied idea [HR, HLM, ...]

Dual approach

Small, tractable

- Manipulate distribution over queries $\mathcal{Q}$
- Optimize: find record in $\mathcal{X}$ with low error

Big, intractable(?)

- Very similar to Dwork, Rothblum, Vadhan

The task

- Sample queries $q_1, \ldots, q_s$ from query distribution (for privacy)

The task

- Sample queries $q_1, \ldots, q_s$ from query distribution (for privacy)
- Pick record minimizing average error over $q_1, \ldots, q_s$:

$$\text{minimize}_r \left\{ (q_1(r) - q_1(D)) + \cdots + (q_s(r) - q_s(D)) \right\}$$

The dual optimization problem

The task

- Sample queries $q_1, \ldots, q_s$ from query distribution (for privacy)
- Pick record minimizing average error over $q_1, \ldots, q_s$:

$$\text{minimize}_r \left\{ (q_1(r) - q_1(D)) + \cdots + (q_s(r) - q_s(D)) \right\}$$

- But D is fixed, so equivalent to:

$$\text{minimize}_r \, q_1(r) + \cdots + q_s(r)$$

The dual optimization problem

The task

- Sample queries $q_1, \ldots, q_s$ from query distribution (for privacy)
- Pick record minimizing average error over $q_1, \ldots, q_s$:

$$\text{minimize}_r \, \{(q_1(r) - q_1(D)) + \cdots + (q_s(r) - q_s(D))\}$$

- But D is fixed, so equivalent to:

$$\text{minimize}_r \, q_1(r) + \cdots + q_s(r)$$

- Pure optimization problem

Theorem

Dual query is (ε, δ)-differentially private.

Privacy, accuracy, and efficiency

Theorem

Dual query is (ε, δ)-differentially private.

Theorem

With high probability, all queries are handled with error α, where

$$\alpha = O\left(\frac{\log|\mathcal{Q}|\log(1/\delta)}{n^{1/3}\varepsilon^{1/3}}\right).$$

Privacy, accuracy, and efficiency

Theorem

Dual query is (ε, δ)-differentially private.

Theorem

With high probability, all queries are handled with error α, where

$$\alpha = O\left(\frac{\log|\mathcal{Q}|\log(1/\delta)}{n^{1/3}\varepsilon^{1/3}}\right).$$

Efficiency

- Optimization problem depends on specific type of queries
- Often this step is hard...

Manage smaller distribution

- Distribution over queries rather than records
- Manageable if $\mathcal{Q}$ not too big

How do we gain?

Manage smaller distribution

- Distribution over queries rather than records
- Manageable if $\mathcal{Q}$ not too big

Optimization

- Intractable, but doesn't involve privacy
- Can use any off-the-shelf solver

How do we gain?

Manage smaller distribution

- Distribution over queries rather than records
- Manageable if $\mathcal{Q}$ not too big

Optimization

- Intractable, but doesn't involve privacy
- Can use any off-the-shelf solver

Further heuristics

- Guarantee privacy, but relax accuracy
- If optimization problem too hard, stop solver early
- Run for fewer rounds

Does it perform well?

Queries and data

- Database with binary attributes
- Randomly generated data, as well as real data
- Three-way marginal queries

Does it perform well?

Queries and data

- Database with binary attributes
- Randomly generated data, as well as real data
- Three-way marginal queries

"What fraction of database satisfies A, B and C?"

Does it perform well?

Queries and data

- Database with binary attributes
- Randomly generated data, as well as real data
- Three-way marginal queries

"What fraction of database satisfies A, B and C?"

Optimization problem

- Related to MAX3SAT
- Encode as integer program and solve with CPLEX
- Take best solution found in 60s

Does it perform well?

Queries and data

- Database with binary attributes
- Randomly generated data, as well as real data
- Three-way marginal queries

"What fraction of database satisfies A, B and C?"

Optimization problem

- Related to MAX3SAT
- Encode as integer program and solve with CPLEX
- Take best solution found in 60s

Hardware

- Medium performance desktop computer

Does it perform well?

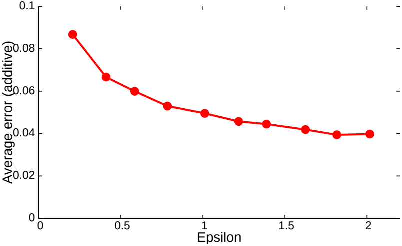


Figure : Average error versus epsilon

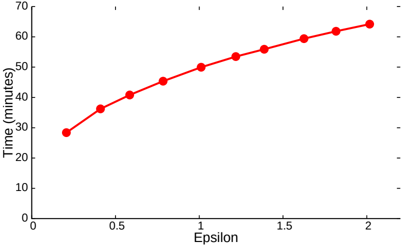


Figure : Runtime versus epsilon

Details

- Real networking data, $\sim$ 100 independent binary attributes
- $\sim$ 500k records, $\sim$ 500k queries
- Most of time spent evaluating queries, rather than optimizing

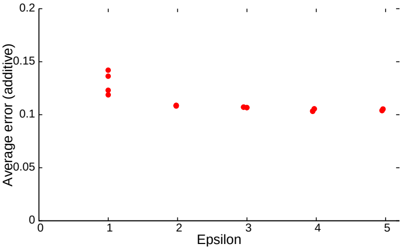


Figure : Average error versus epsilon

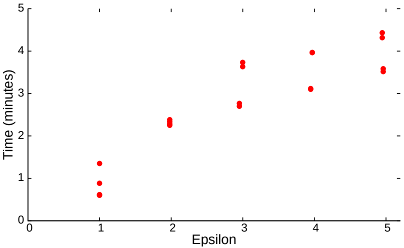


Figure : Runtime versus epsilon

Details

- Randomly biased data, 2000 independent binary attributes
- 100k records, 100k queries

Wrapping up

More practical query release

- Handle higher dimensional data
- Use standard solvers on the hard step
- Performs well in practice

Wrapping up

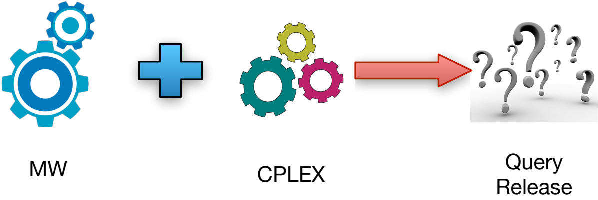
More practical query release

- Handle higher dimensional data
- Use standard solvers on the hard step
- Performs well in practice

Ongoing/future work

- More experiments, solvers. Scaling?
- Other classes of queries?
- When is the optimization problem easy?
- Other heuristics for privacy?

Dual Query Release



Marco Gaboardi, Emilio Gallego Jésus Arias,
Justin Hsu, Aaron Roth, Steven Wu

University of Pennsylvania

December 11th, 2013