CSE660 Differential Privacy

October 16, 2017

Marco Gaboardi
Room: 338-B
gaboardi@buffalo.edu
http://www.buffalo.edu/~gaboardi

(ε,δ)-Differential Privacy

Definition

Given $\varepsilon, \delta \geq 0$, a probabilistic query $Q: X^n \to R$ is (ε,δ)-differentially private iff for all adjacent database b_1, b_2 and for every $S \subseteq R$:

$$\Pr[Q(b_1) \in S] \leq \exp(\varepsilon)\Pr[Q(b_2) \in S] + \delta$$

Multiple queries

Question: how much perturbation do we have if we want to answer n counting queries with Laplace under ε-DP?

Multiple queries

Question: how much perturbation do we have if we want to answer n counting queries with Laplace under ε-DP?

Using standard composition we have as a max error

$$O\Big(\frac{n}{\epsilon_{\text{global}} n}\Big) = O\Big(\frac{1}{\epsilon_{\text{global}}}\Big)$$

Notice that if we don't renormalize this is of the order of

$$O\Big(\frac{n}{\epsilon_{\text{global}}}\Big)$$

bigger than the sample error.

Advanced Composition

Question: how much perturbation do we have if we want to answer n queries under (ε,δ)-DP?

Using advanced composition we have as a max error

$$O\left(\frac{1}{\epsilon_{\text{global}}\sqrt{n}}\right)$$

If we don't renormalize this is of the order of

$$O\left(\frac{\sqrt{n}}{\epsilon_{\text{global}}}\right)$$

comparable to the sample error.

[DworkRothblumVadhan10, SteinkeUllman16]

Answering multiple queries

We have seen several methods to answer a single query:
- Randomized Response
- Laplace Mechanism
- Exponential Mechanism

And methods to answer multiple queries with small error:
- Standard composition - we can answer $\sqrt{n}$ queries.
- Advanced composition - we can answer n queries.

Question: Can we do better?

SmallDB: Answering multiple linear queries

Algorithm 5 Pseudo-code for SmallDB

1: **function** SmallDB(D, Q, ϵ, α)
2: Let $m = \frac{\log |Q|}{\alpha^2}$
3: Let $u : \mathcal{X}^n \times \mathcal{X}^m \to \mathbb{R}$ be defined as:

$$u(D, D_i) = -\max_{q \in Q} |q(D) - q(D_i)|$$

4: Let $D' \leftarrow \mathcal{M}_E(D, u, \epsilon)$
5: **return** D'
6: **end function**

SmallDB: Answering multiple linear queries

Equivalently, for any database x with

$$\|x\|_1 \geq \frac{16 \log |\mathcal{X}| \log |\mathcal{Q}| + 4 \log\left(\frac{1}{\beta}\right)}{\varepsilon \alpha^3}$$

with probability $1 - \beta$: $\max_{f \in \mathcal{Q}} |f(x) - f(y)| \leq \alpha$.

Answering multiple queries

We have seen several methods to answer a single query:

- Randomized Response
- Laplace Mechanism
- Exponential Mechanism

And methods to answer multiple queries with small error:

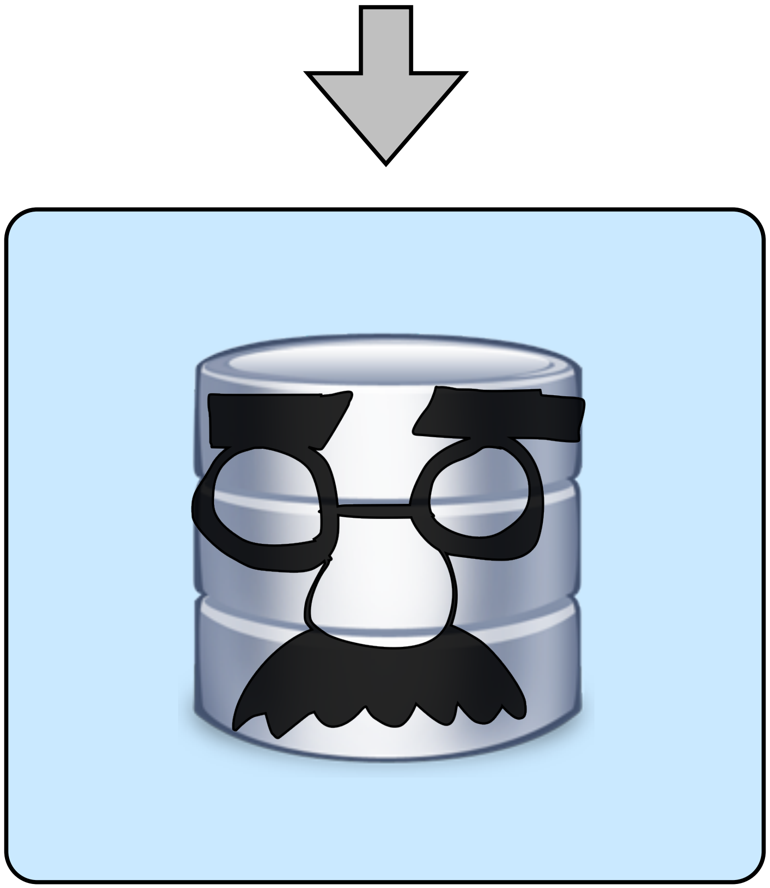
- Standard composition - we can answer $\sqrt{n}$ queries.
- Advanced composition - we can answer n queries.

If we allow coordinating noise among different queries we can answer an exponential number of queries.

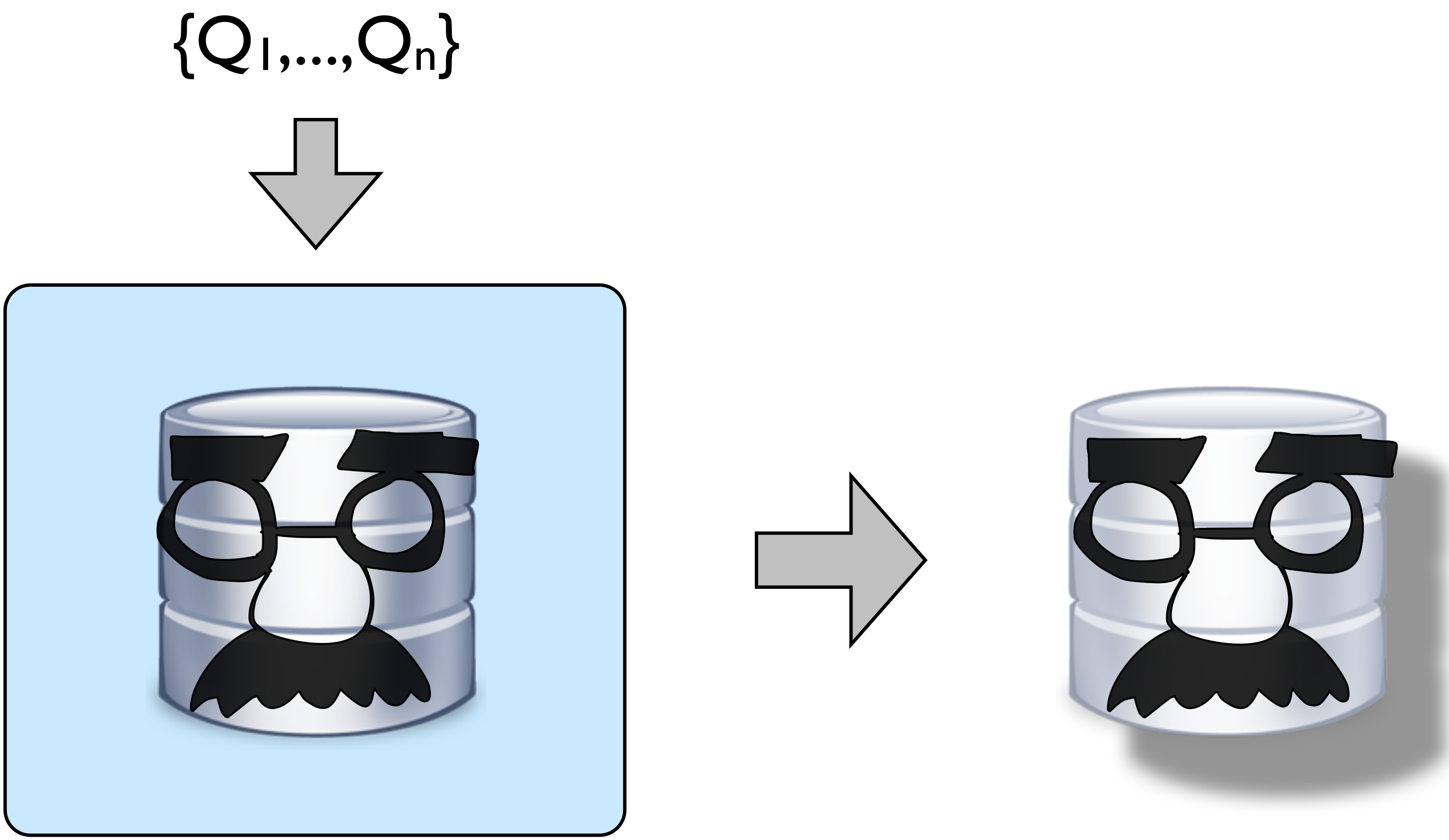
Data Release: IDC

Data Release: IDC

$\{Q_1,\ldots,Q_n\}$



Data Release: IDC



Iterative Database Construction

Given a database D, a set of queries $\{Q_1,\ldots,Q_n\}$ and a target accuracy α it generates a sequence $D_1,\ldots,D_n$ of synthetic DBs such that:

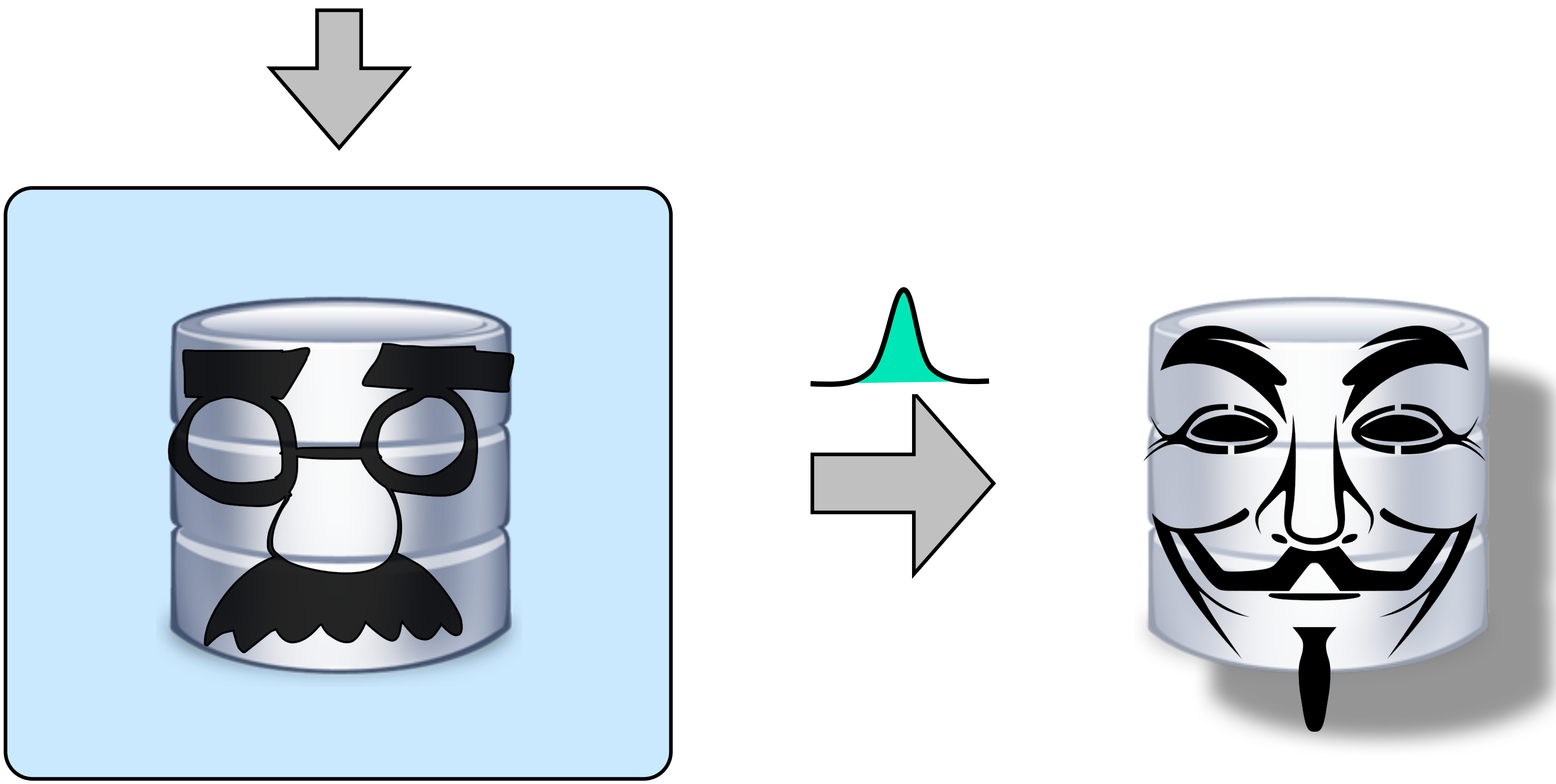
- it gives increasingly better approximations of D,
- D_{t+1} is generated by D_t using only one query Q_t maximizing the difference:

$$| Q_t(D) - Q_t(D_t) | \geqq k$$

- D_n satisfies the target accuracy α.

Private Data Release

$\{Q_1, \ldots, Q_n\}$



Private Iterative Database Construction

Given a database D, a set of queries $\{Q_1,\ldots,Q_n\}$ and a target accuracy α it generates a sequence $D_1,\ldots,D_n$ of synthetic DBs such that:

- it gives increasingly better approximations of D,
- D_{t+1} is privately generated by D_t using only one query $\mathcal{Q}_t = Q_t + \texttt{noise}$ maximizing the difference:

$$| Q_t(D) - \mathcal{Q}_t(D_t) | \geqq k$$

- D_n satisfies the target accuracy α.

MWEM

Multiplicative Weight Exponential Mechanism

An algorithm for IDC for linear queries based on:
- the Exponential mechanism to select the query maximizing the difference,
- the Multiplicative Weight update rule to update the database.

Dataset as a distribution over the universe

The algorithm views a dataset D as a distribution over rows $x \in \mathcal{X}$.

$$D(x) = \frac{\#\{i \in [n] \, : \, d_i = x\}}{n}$$

Then,

$$q(D) = \mathbb{E}_{x \to D}[q(x)]$$

We denote by D_0 the uniform distribution over $\mathcal{X}$.

D_0	1/n	1/n	1/n	1/n	1/n	1/n	.....	.....	.....	1/n

MWEM

Algorithm 9 Pseudo-code for MWEM

function MWEM(D, $q_1, \ldots, q_m$, T, ϵ, D_0)

  for $i \leftarrow 1, \ldots, T$ **do**

    $u_i(D, q) = |q(D_{i-1}) - q(D)|$

    $\hat{q} \leftarrow \mathsf{ExpMech}(D, u_i, n\epsilon/2T)$

    $m_i \leftarrow \hat{q}(D) + \mathsf{Lap}(T/n\epsilon)$

    $D_i(x) = D_{i-1}(x) \times \mathsf{exp}(\hat{q}(x)\frac{(m - \hat{q}(D_{i-1}))}{2})$

    $D_i = \text{renormalize}(D_i)$

  end for

  return $\mathsf{avg}_{i<T} D_i$

end function

Update part using the MW algorithm and a threshold obtained with Laplace.

MW Intuition

$$D_i(x) = D_{i-1}(x) \times \text{exp}(q(x)\frac{(q(D)-q(D_{i-1}))}{2n})$$

For a given query q:

- If $q(D) \gg q(D_{i-1})$, we should scale up the weights on records contributing *positively*, and scale down the ones contributing *negatively*,
- If $q(D) \ll q(D_{i-1})$, we should scale down the weights on records contributing *positively*, and scale up the ones contributing *negatively*.

MW Intuition

Let's consider counting queries.

D_0	1/n	1/n	1/n	1/n	1/n	1/n	.....	.....	.....	1/n
	$\hat{q}_1(x_1)$	$\hat{q}_1(x_2)$	$\hat{q}_1(x_3)$	$\hat{q}_1(x_4)$	$\hat{q}_1(x_5)$	$\hat{q}_1(x_6)$	... ...	... ...	... ...	$\hat{q}_1(x_{\|\mathcal{X}\|})$
	0	1	1	0	1	0	.....	.....	.....	1
	↓	↑	↑	↓	↑	↓	....	....	....	↑
D_1	1/n	$e^{1/2n}$/n	$e^{1/2n}$/n	1/n	$e^{1/2n}$/n	1/n	.....	.....	.....	$e^{1/2n}$/n

Let's assume $\hat{q}_1(D) - \hat{q}(D_0) = 1$ then $D_1(x_j) = D_0(x_j) \times \exp(\frac{q_1(x_j)}{2n})$

MW Intuition

Let's consider counting queries.

D_0

1/n	1/n	1/n	1/n	1/n	1/n	.....	.....	.....	1/n
$\hat{q}_1(x_1)$	$\hat{q}_1(x_2)$	$\hat{q}_1(x_3)$	$\hat{q}_1(x_4)$	$\hat{q}_1(x_5)$	$\hat{q}_1(x_6)$	... ...	... ...	... ...	$\hat{q}_1(x_{\lvert\mathcal{X}\rvert})$
0	1	1	0	1	0	.....	.....	.....	1

D_1

1/n	$1/e^{1/2n}n$	$1/e^{1/2n}n$	1/n	$1/e^{1/2n}n$	1/n	.....	.....	.....	$1/e^{1/2n}n$

Let's assume $\hat{q}_1(D) - \hat{q}(D_0) = -1$ then $D_1(x_j) = D_0(x_j) \times \exp(\frac{-q_1(x_j)}{2n})$

MWEM

Algorithm 9 Pseudo-code for MWEM

function MWEM(D, $q_1, \ldots, q_m$, T, ϵ, D_0)
    for $i \leftarrow 1, \ldots, T$ **do**
        $u_i(D, q) = |q(D_{i-1}) - q(D)|$
        $\hat{q} \leftarrow \mathsf{ExpMech}(D, u_i, n\epsilon/2T)$
        $m_i \leftarrow \hat{q}(D) + \mathsf{Lap}(T/n\epsilon)$
        $D_i(x) = D_{i-1}(x) \times \mathsf{exp}(\hat{q}(x)\frac{(m - \hat{q}(D_{i-1}))}{2})$
        $D_i = \text{renormalize}(D_i)$
    end for
    return $\mathsf{avg}_{i<T} D_i$
end function

MWEM - Privacy

Theorem 1.13. MWEM satisfies ϵ-differnetial privacy.

MWEM

Accuracy Theorem: MWEM achieves max-error:

$$\alpha = O\left(\frac{\sqrt{\log|\mathcal{X}| \cdot \log(1/\delta)} \cdot \boxed{\log|\mathcal{Q}|}}{\varepsilon n}\right)^{1/2}.$$

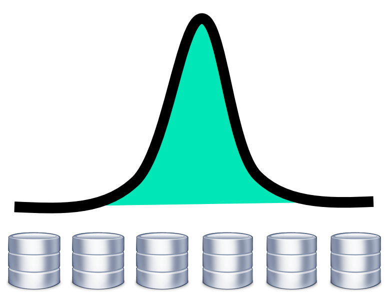
Accuracy Theorem: SmallDB achieves max-error:

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

MWEM

Keep a distribution over the databases, and search for a query which maximize the error, to be used in the update rule.

Distribution



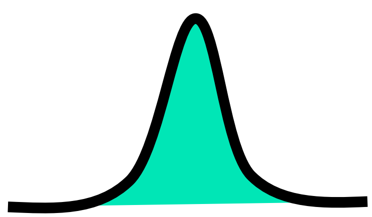
Search space

$\{Q_1,\ldots,Q_n\}$

A dual approach

Keep a distribution over the queries, and search for a record which maximize the error, to be used in the update rule.

Distribution



$\{Q_1,\ldots,Q_n\}$

Search space

Distribution over queries

In general we want to consider a set Q of queries $q_1,\ldots,q_k$ closed under negation.

We denote by Q_0 the uniform distribution over Q:

Q_1										
	1/|Q|	1/|Q|	1/|Q|	1/|Q|	1/|Q|	1/|Q|	…..	…..	…..	1/|Q|

DualQuery

Algorithm 11 Pseudo-code for DualQuery

function DUALQUERY(D, Q, C, s, α, Q_0)
  for $i \leftarrow 1, \ldots, C$ **do**
    Sample s queries $q_1^s, \ldots, q_s^s$ from Q
    Find x_i such that
      $\left(\frac{1}{s}\sum_j q_j^s(x_i)\right) \geq \left(\max_x \frac{1}{s}\sum_j q_j^s(x)\right) - \alpha/4$
    $Q_i(q) = Q_{i-1}(q) \times \mathsf{exp}(-\alpha(\frac{q(x_i)-q(D))}{2n})$
    $Q_i = \text{renormalize}(Q_i)$
  end for
  return $\bigcup_{i<C} x_i$
end function

DualQuery - Privacy

Theorem 1.15. DualQuery is differentially private for:

$$\epsilon = \frac{\alpha T(T-1)s}{4n}$$

DualQuery revisited

The privacy proof follows by composition

Algorithm 11 Pseudo-code for DualQuery

function DUALQUERY(D, Q, T, s, α, Q_1)
    Sample s queries $q_1^s, \ldots, q_s^s$ from Q
    Find x_1 such that
        $\left(\frac{1}{s}\sum_j q_j^s(x_1)\right) \geq \left(\max_x \frac{1}{s}\sum_j q_j^s(x)\right) - \alpha/4$
    for $i \leftarrow 2, \ldots, T$ **do**
        $u_i(D, q) = \sum_j^{i-1} q(x_j) - q(D)$
        Sample s queries $q_1^s, \ldots, q_s^s$ as
            $q_k^s \leftarrow \mathsf{ExpMech}(D, u_i, \frac{\alpha(i-1)}{n})$
        Find x_i such that
            $\left(\frac{1}{s}\sum_j q_j^s(x_i)\right) \geq \left(\max_x \frac{1}{s}\sum_j q_j^s(x)\right) - \alpha/4$
    end for
    return $\bigcup_{i \leq T} x_i$
end function

DualQuery - Accuracy

Accuracy Theorem: DualQuery achieves max-error:

$$\alpha = O\left(\frac{\log^{1/2}|\mathcal{Q}|\log^{1/6}(1/\delta)\log^{1/6}(2|\mathcal{X}|/\gamma)}{n^{1/3}\varepsilon^{1/3}}\right)$$

Accuracy Theorem: SmallDB achieves max-error:

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

Accuracy Theorem: MWEM achieves max-error:

$$\alpha = O\left(\frac{\sqrt{\log|\mathcal{X}|\cdot\log(1/\delta)}\cdot\log|\mathcal{Q}|}{\varepsilon n}\right)^{1/2}.$$

DualQuery novelty?

Algorithm 11 Pseudo-code for DualQuery

function DUALQUERY(D, Q, T, s, α, Q_1)
    Sample s queries $q_1^s, \ldots, q_s^s$ from Q
    Find x_1 such that
        $\left(\frac{1}{s}\sum_j q_j^s(x_1)\right) \geq \left(\max_x \frac{1}{s}\sum_j q_j^s(x)\right) - \alpha/4$
    for $i \leftarrow 2, \ldots, T$ **do**
        $u_i(D, q) = \sum_j^{i-1} q(x_j) - q(D)$
        Sample s queries $q_1^s, \ldots, q_s^s$ as
            $q_k^s \leftarrow \mathsf{ExpMech}(D, u_i, \frac{\alpha(i-1)}{n})$
        Find x_i such that
            $\left(\frac{1}{s}\sum_j q_j^s(x_i)\right) \geq \left(\max_x \frac{1}{s}\sum_j q_j^s(x)\right) - \alpha/4$
    end for
    return $\bigcup_{i \leq T} x_i$
end function

The most expensive task is non-private

We can use standard optimization tools

Example k-way marginals

Let's consider the universe domain $\mathcal{X} = \{0,1\}^d$ and let's consider $\vec{v} \in \{1, \bar{1}, \ldots, d, \vec{d}\}^k$ with $1 \leq k \leq d$ and

$$q_{\vec{v}}(x) = q_{v_1}(x) \wedge q_{v_1}(x) \wedge \cdots \wedge q_{v_k}(x)$$

where $q_j(x) = x_j$ and $q_{\bar{j}}(x) = \neg x_j$

We call a conjunction or k-way marginal the associated counting query

$$q_{\vec{v}} : \mathcal{X}^n \to [0,1]$$

We can create a corresponding integer program problem.

Example 3-way marginals

$$\max \sum_i c_i + \sum_j d_j$$

$$\text{with } \forall \widehat{u_i} = q_{abc} : \; x_a + x_b + x_c \geq 3c_i$$

$$\forall \widehat{v_j} = \overline{q_{abc}} : \; (1 - x_a) + (1 - x_b) + (1 - x_c) \geq d_j$$

$$x_i, c_i, d_i \in \{0, 1\}$$