Faster Algorithms for Privately Releasing Marginals

Justin Thaler (Harvard University)
Jon Ullman (Harvard University)
Salil Vadhan (Harvard University)

K-way Marginal Queries

$D \in (\{0, 1\}^d)^n$

Exercise?	Healthy?	Ice Cream?	Criminal?
Y	Y	Y	Y
N	N	N	N
Y	N	Y	N
Y	N	Y	Y

Query on a row: $q(x) = \text{Ice Cream?} \wedge \text{Criminal?}$

Query on database: $(1/n) \sum_i q(x_i)$

- **k-way marginal queries**: q has at most k literals.
- Number of k-way marginal queries $\sim d^k$.

Goal: Private One-Shot Release Mechanism

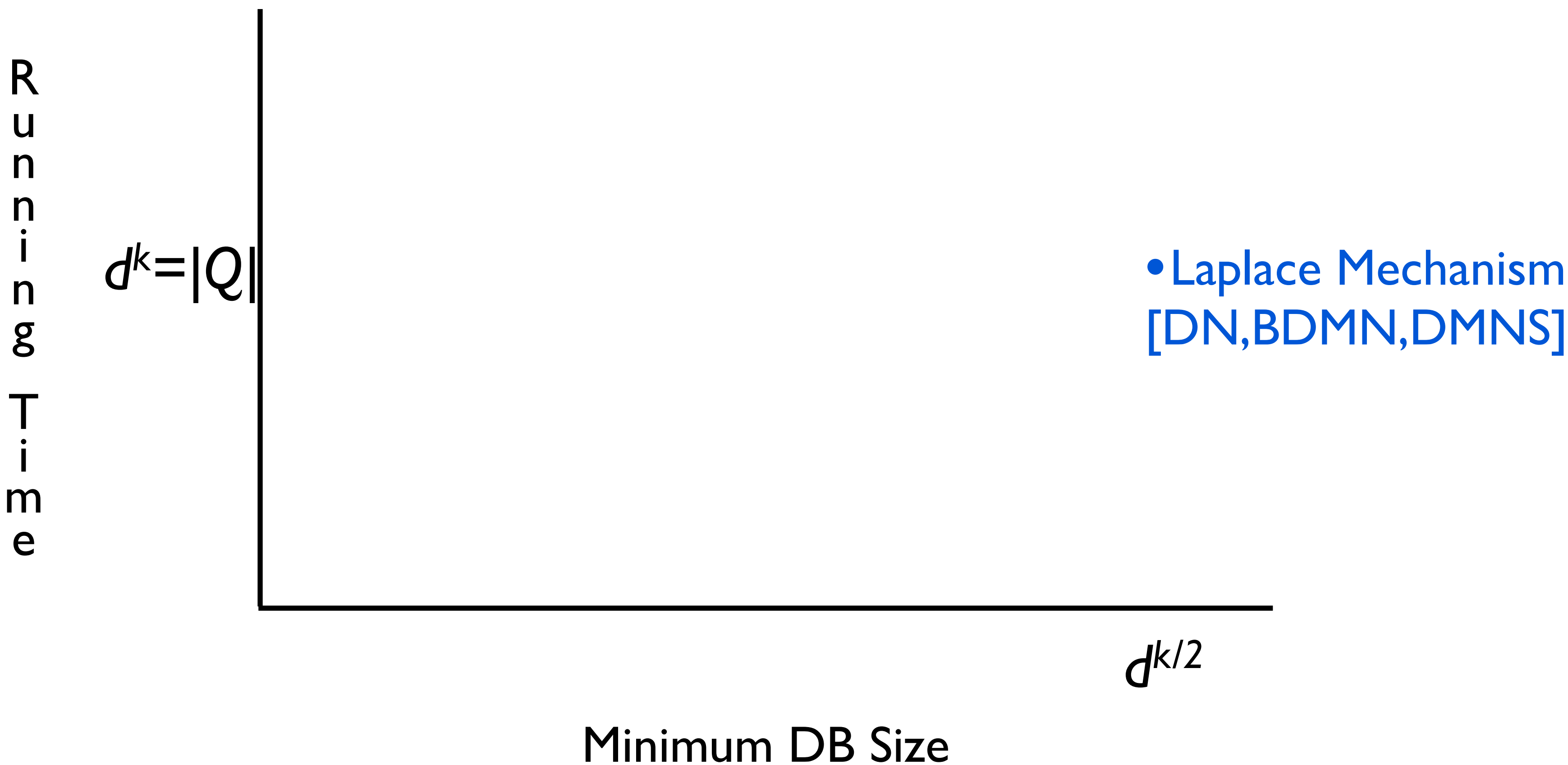
- Want to release a *summary* of D such that for all k-way marginals q:

$$|\text{Summary}(q) - q(D)| \leq .01$$

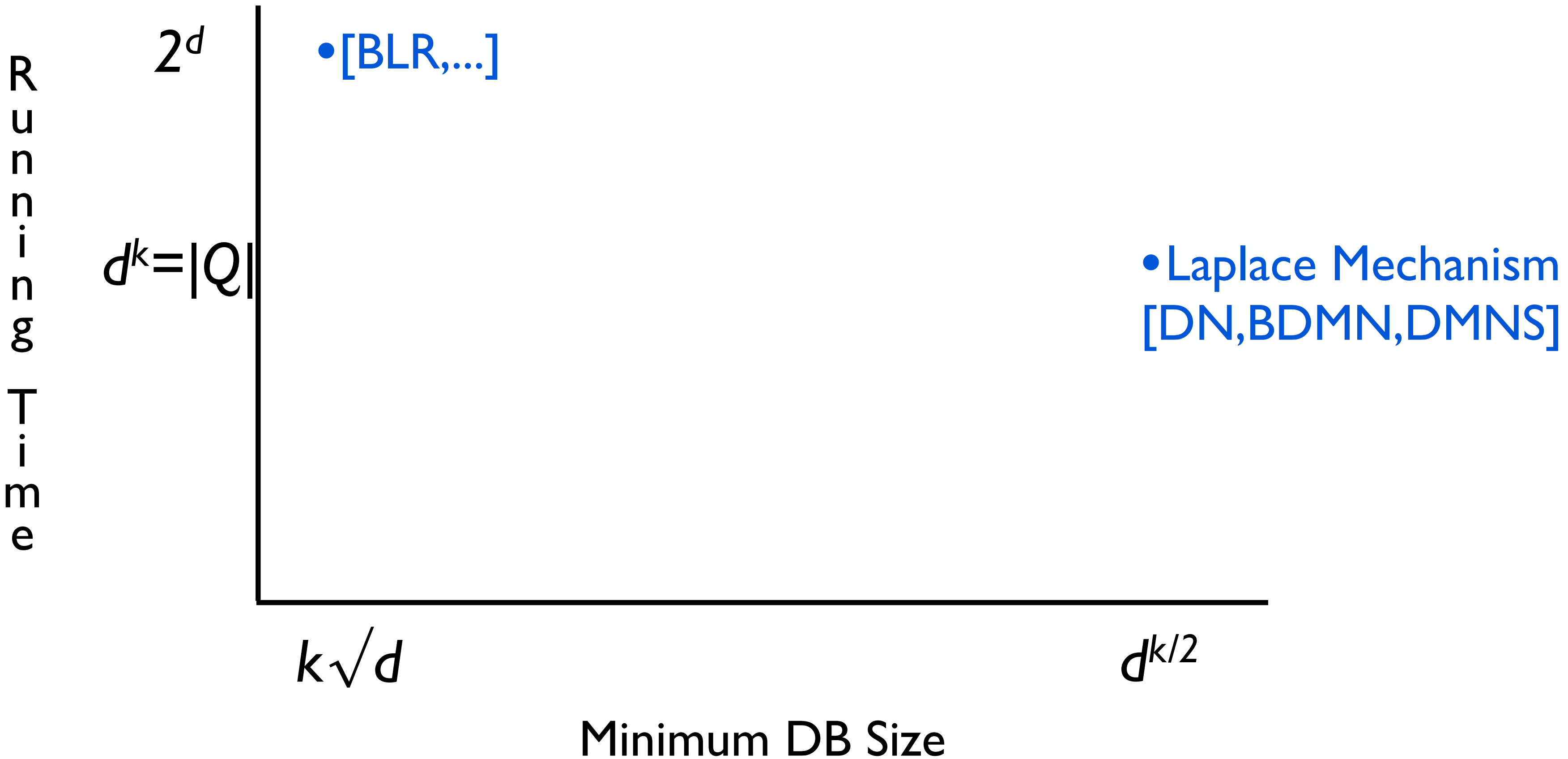
- Two parameters to optimize: running time of the sanitizer and minimal database size required.

Prior Work on Marginals

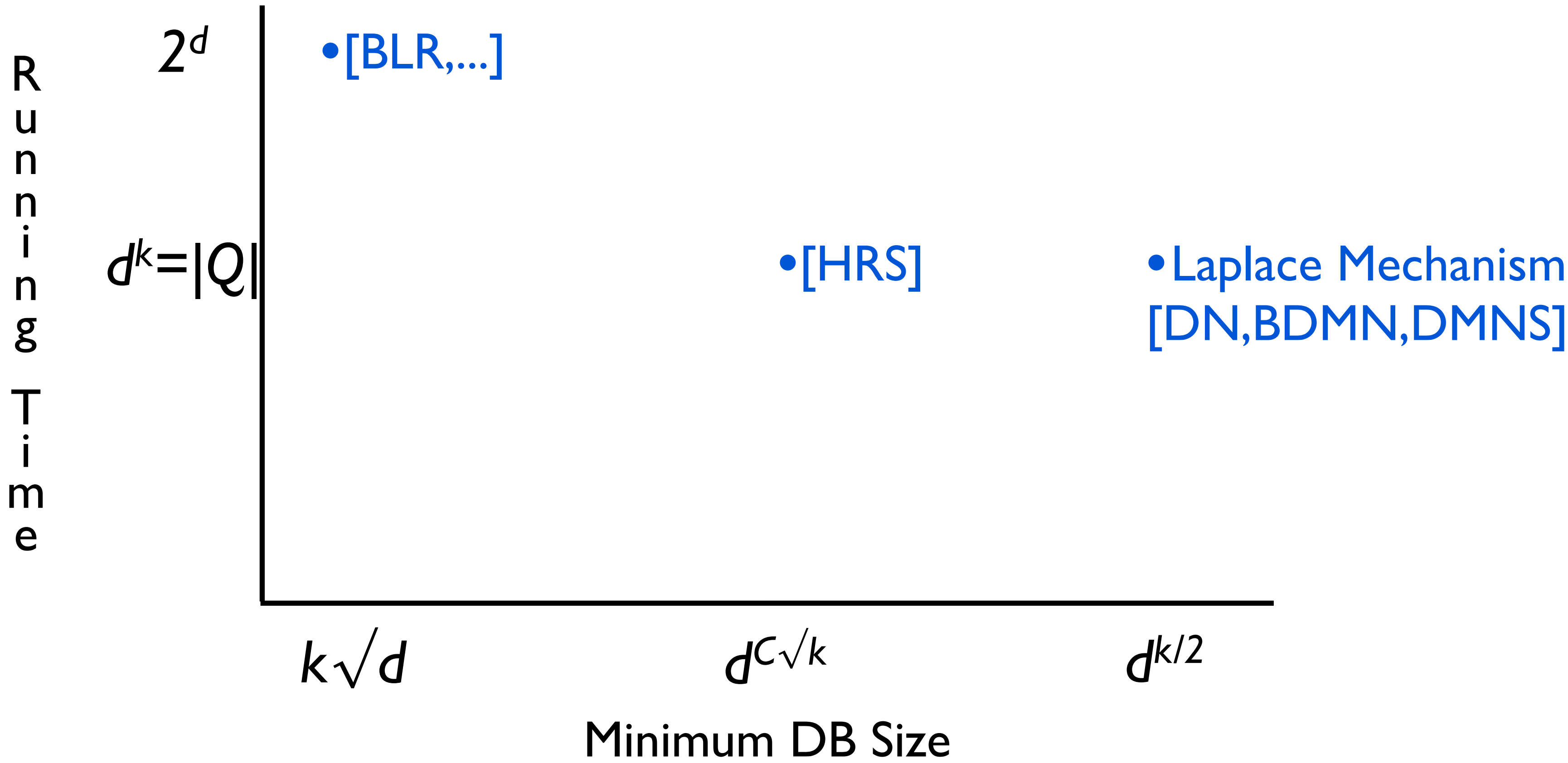
Prior Work on Marginals



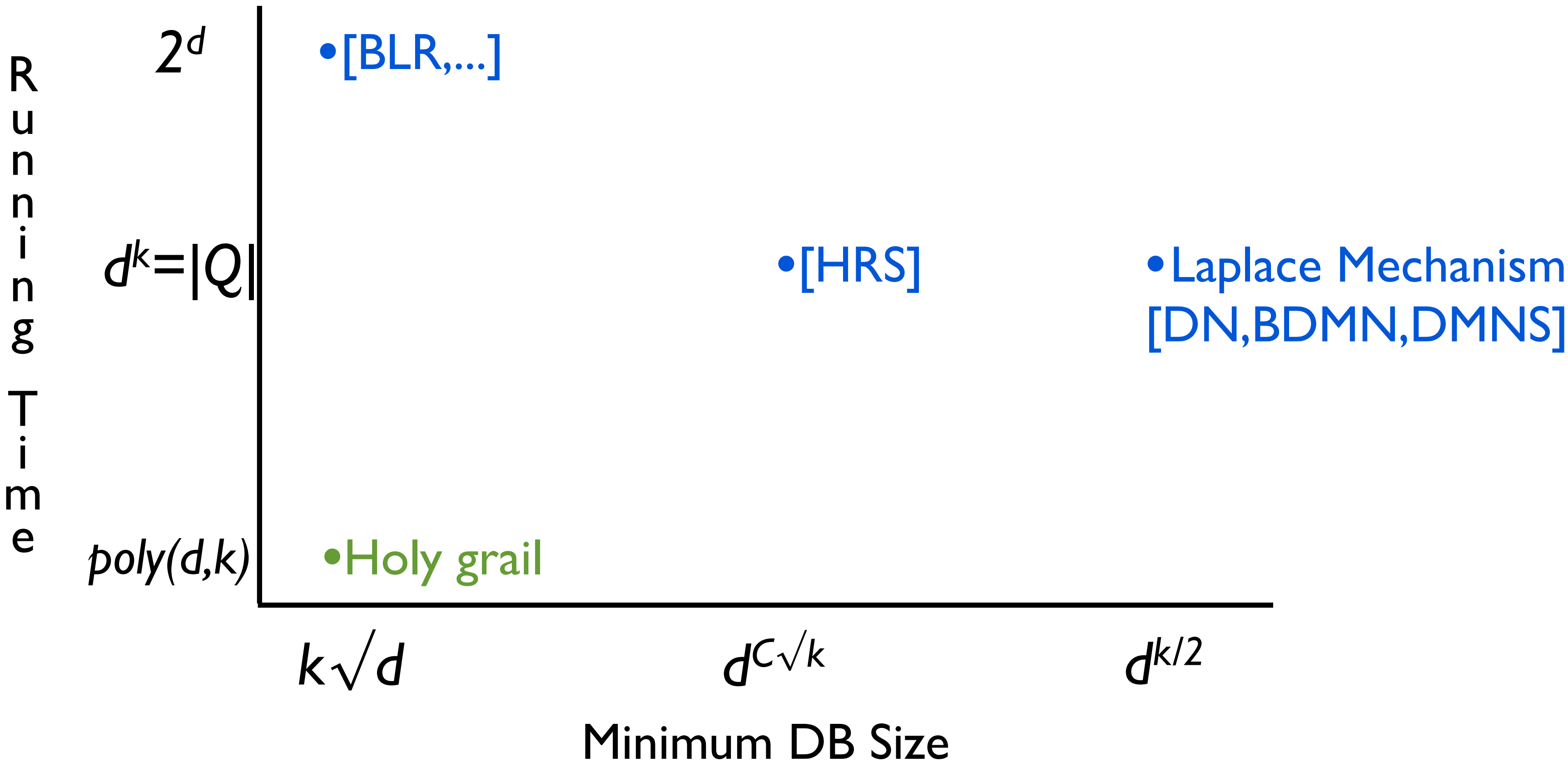
Prior Work on Marginals



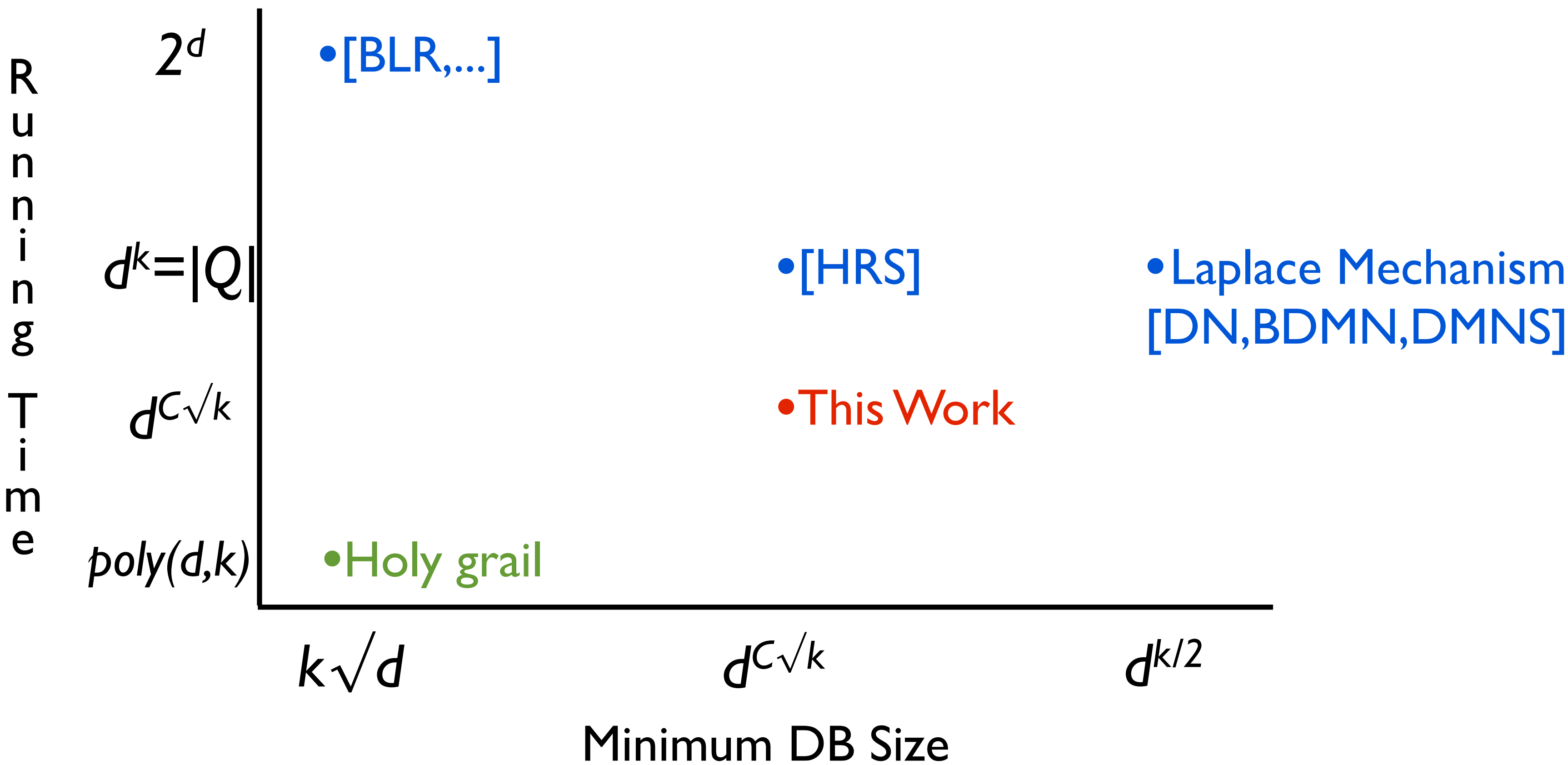
Prior Work on Marginals



Prior Work on Marginals



Our Result



Running Time
2^d
$d^k = |Q|$
$d^{C\sqrt{k}}$
$poly(d,k)$
•[BLR,...]
•[HRS]
•Laplace Mechanism [DN,BDMN,DMNS]
•This Work
•Holy grail
$k\sqrt{d}$
$d^{C\sqrt{k}}$
$d^{k/2}$
Minimum DB Size

Our Results

- Faster algorithm for privately releasing marginals with small worst-case error (accuracy $\pm.01$).
  - Time: $d^{C\sqrt{k}}$, minimum database size: $n \geq d^{C\sqrt{k}}$.
  - First sanitizer for k-way marginals with running time and minimal DB size sublinear in total number of k-way marginals $\sim d^k$.
  - Can handle more general settings as well (e.g. where rows of the DB represent *decision lists*).

Our Algorithm

$$D \in (\{0, 1\}^d)^n$$

$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	$\mathbf{x_4}$
1	1	1	0
0	1	0	0
0	0	1	1
0	0	0	1

- View each row x as a function f_x from queries to $\{0, 1\}$:
  $f_x(q) = 1$ iff row x satisfies marginal q.

Our Algorithm

$$D \in (\{0, 1\}^d)^n$$

x_1	x_2	x_3	x_4
1	1	1	0
0	1	0	0
0	0	1	1
0	0	0	1

- View each row x as a function f_x from queries to $\{0, 1\}$:
  $f_x(q) = 1$ iff row x satisfies marginal q.
- For every x, there exists a d-variate polynomial p_x such that:
  - $|p_x(q)-f_x(q)| \leq .01$ for all q corresponding to k-way marginals.
  - Degree(p) $\leq C\sqrt{k}$ for some constant C.
  - All coefficients of p are in $[\pm d^{C\sqrt{k}}]$.

Our Algorithm

$$D \in (\{0, 1\}^d)^n$$

$p_{x1}(y) = 3y_1y_2+7y_2y_4+\ldots$

$p_{x2}(y) = 4y_1y_2-3y_2y_4+\ldots$

$p_{x3}(y) = -3y_1y_2+2y_2y_4+\ldots$

$p_{x4}(y) = 8y_1y_2+y_2y_4+\ldots$

x_1	x_2	x_3	x_4
1	1	1	0
0	1	0	0
0	0	1	1
0	0	0	1

Our Algorithm

$$D \in (\{0, 1\}^d)^n$$

$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	$\mathbf{x_4}$
1	1	1	0
0	1	0	0
0	0	1	1
0	0	0	1

$p_{x1}(y) = 3y_1y_2+7y_2y_4+\ldots$

$p_{x2}(y) = 4y_1y_2-3y_2y_4+\ldots$

$p_{x3}(y) = -3y_1y_2+2y_2y_4+\ldots$

$p_{x4}(y) = 8y_1y_2+y_2y_4+\ldots$

- Let $p_D(y) = (1/n) \Sigma_i p_{xi}(y)$ be the average of the polynomials approximating each row.
- We output a noisy version of p_D.

Our Algorithm

$$D \in (\{0, 1\}^d)^n$$

x_1	x_2	x_3	x_4
1	1	1	0
0	1	0	0
0	0	1	1
0	0	0	1

$p_{x1}(y) = 3y_1y_2 + 7y_2y_4 + \ldots$

$p_{x2}(y) = 4y_1y_2 - 3y_2y_4 + \ldots$

$p_{x3}(y) = -3y_1y_2 + 2y_2y_4 + \ldots$

$p_{x4}(y) = 8y_1y_2 + y_2y_4 + \ldots$

- Let $p_D(y) = (1/n)\sum_i p_{xi}(y)$ be the average of the polynomials approximating each row.
- We output a noisy version of p_D.
  - Degree$(p_D) = C\sqrt{k}$. So about $d^{C\sqrt{k}}$ coefficients.
  - p_D has coefficients in $[\pm d^{C\sqrt{k}}]$, each coeff has sensitivity $\sim d^{C\sqrt{k}}/n$
  - Add independent Laplace noise to each coeff of magnitude $\sim d^{C\sqrt{k}}/n$.

Conclusion

- Previous sanitizers [HRS, etc.] gave a learning algorithm restricted access to the DB.
- We cut out the learning algorithm, and give our sanitizer direct access to the database.
  - We use the same structural results results underlying many learning algorithms.
- Does relying on learning algorithms for differential privacy unnecessarily tie our hands?