Algorithms for Answering Linear Queries
Part II

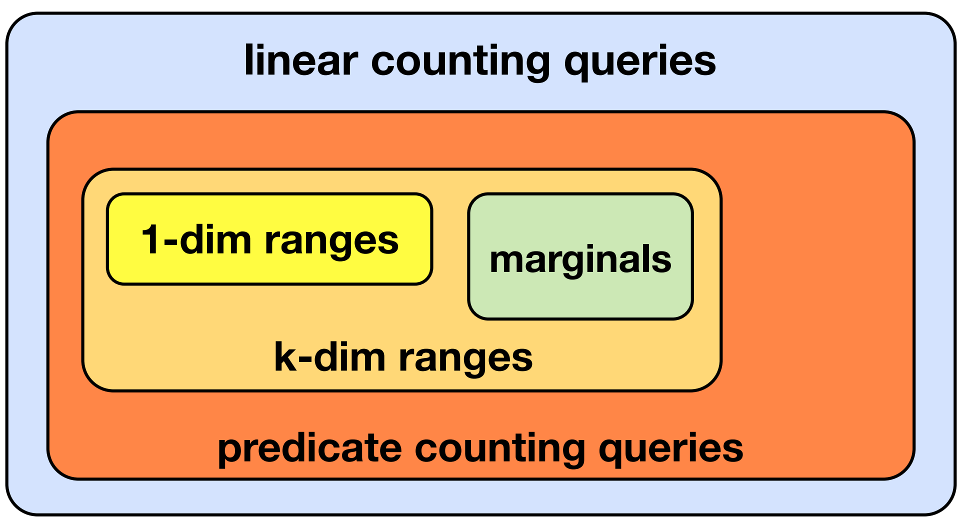
Sasho Nikolov (U Toronto)

Gerome Miklau (*Univ. of Massachusetts, Amherst)*
Ryan McKenna (*Univ. of Massachusetts, Amherst)*

Jan 30, 2019 — Simons Institute

Task: batch (non-interactive) query answering

- **Answer:** a fixed set of **linear counting queries**
    - complex data analysis task into simpler queries.
    - multiple users each issuing one or more queries.
    - uncertainty about the eventual query answers needed--design workload to include all queries possibly of interest.

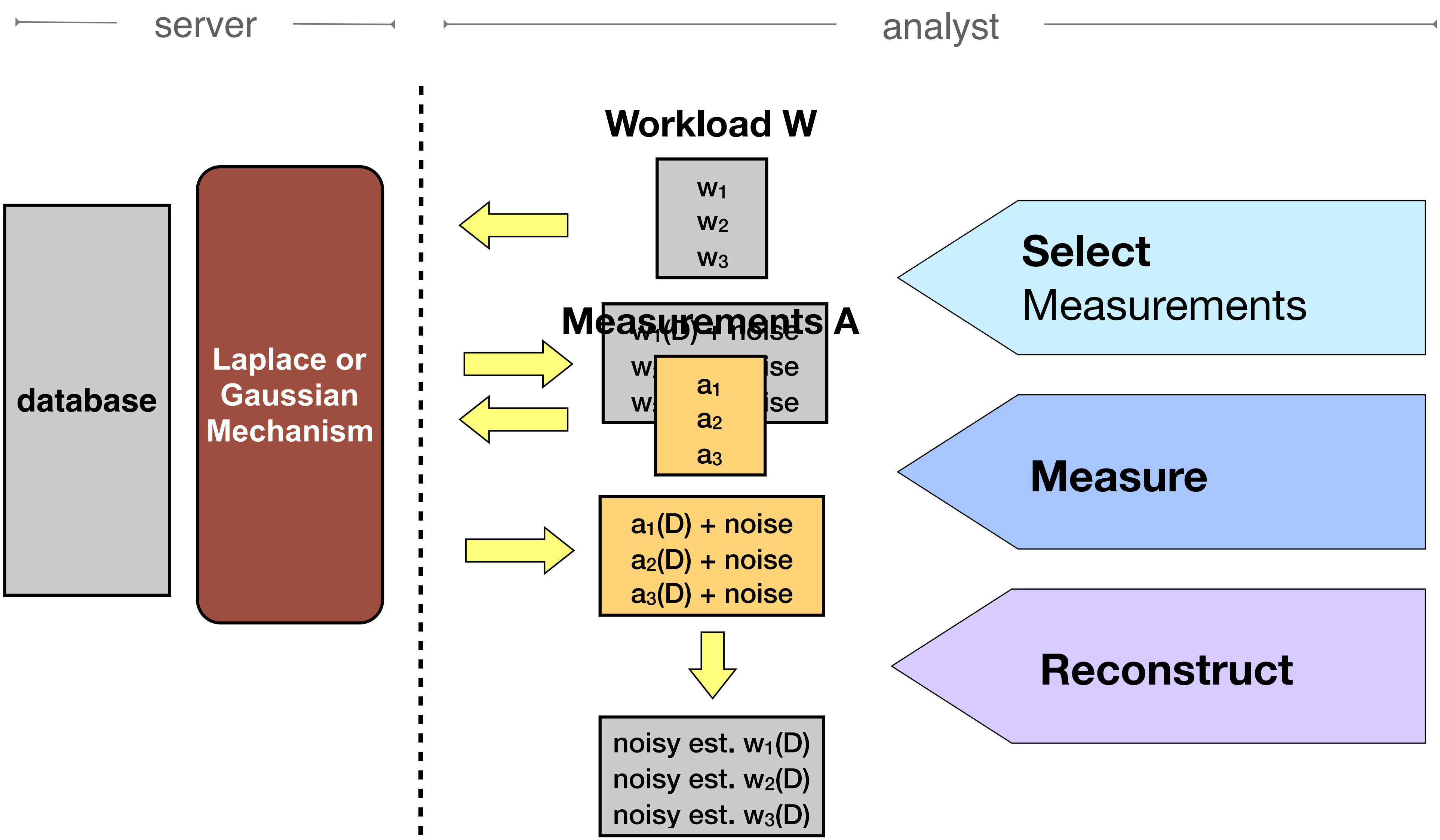


Outline

1. **Algorithm landscape**
2. Motivating challenge: a Census workload
3. Scaling the matrix mechanism
4. Results on the Census workload
5. Data-adaptive algorithms and trade-offs
6. Open problems

Break ?

Approach 1: data-agnostic mechanisms



Data-agnostic mechanisms

- Many algorithms belong to the select-measure-reconstruct paradigm, which adapt measurements to the workload

Workload	Strategy (Measurements)		Citation
any	Fixed	Identity	[Dwork, TCC '06]
low-order marginals		Fourier basis queries	[Barak, PODS '07]
all one-dim range queries		Hierarchical ranges	[Hay, PVLDB '10]
all (multi-dim) range queries		Haar wavelet queries	[Xiao, ICDE '10]
2-dim range queries		Quad-tree queries	[Cormode, ICDE '12]
set of linear queries	Optimized	set of linear queries	[Li, PODS '10] [Li, PVLDB '12]
sets of data cubes		sets of data cubes	[Ding, SIGMOD '11]
set of linear queries		set of linear queries	[Yuan, VLDB '12]
range queries		hierarchical ranges	[Qardaji, PVLDB '13]
range queries		weighted hierarchical ranges	[Li, VLDB '14]

Selected measurements for range queries

Given workload W of range queries:

Measurement Set A	**Resulting mechanism**
A = Identity matrix	a common baseline
A = Haar wavelet	**[Xiao, ICDE ‘10]**
A = tree based	**[Hay, PVLDB ‘10] [Bolot 2011]** **[Cormode, ICDE ’12] [Qardaji, PVLDB ’13]**

Strategy matrices for **1D range queries**

(for a domain of size 4)

Identity

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{I}$

Hierarchical

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{H}$

Wavelet

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

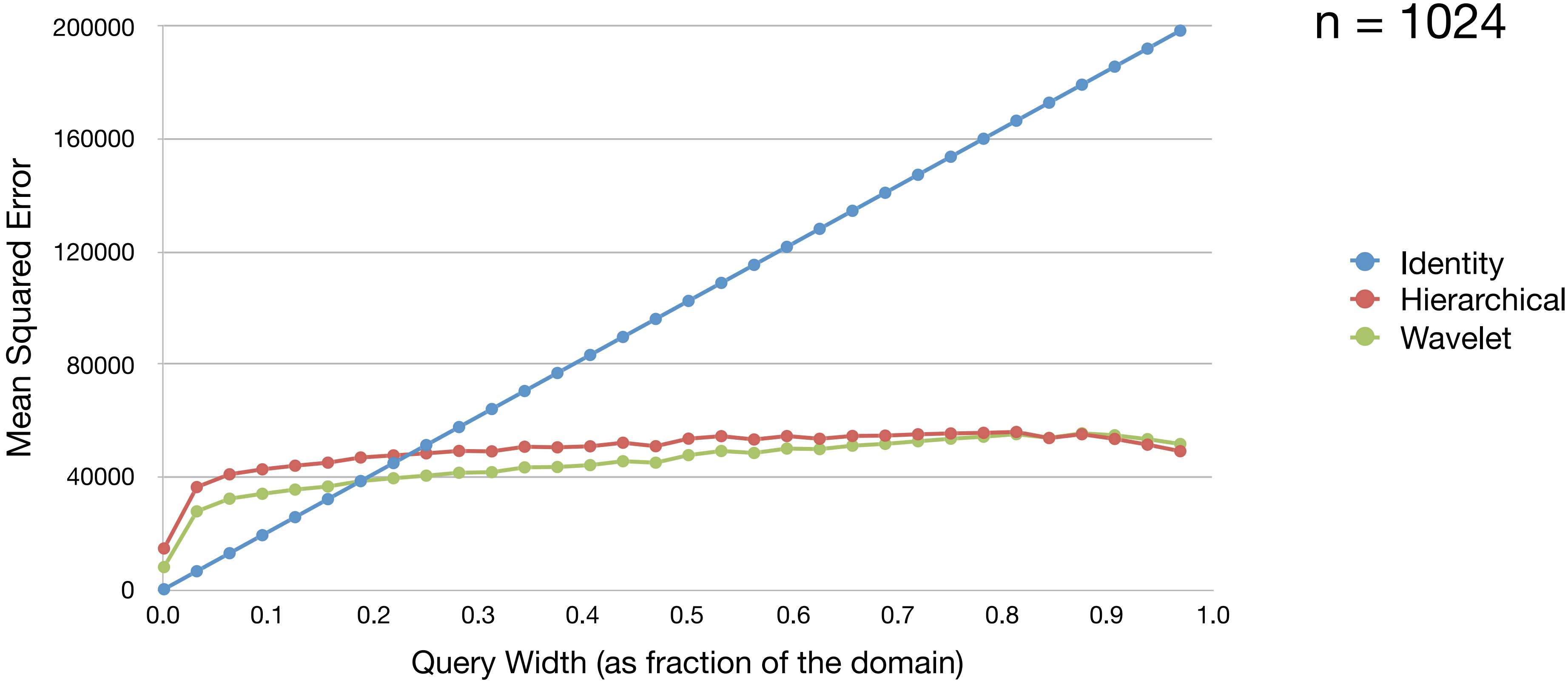
$\mathbf{Y}$

A good strategy has **low sensitivity** but permits **low-error reconstruction** of the workload queries.

Error: workload of all range queries

ε-differential privacy

ε = 0.1
n = 1024



Strategy matrices equivalent to wavelet

$$
\begin{array}{|c|c|c|c|}
\hline
1 & 1 & 1 & 1 \\ \hline
1 & 1 & -1 & -1 \\ \hline
1 & -1 & 0 & 0 \\ \hline
0 & 0 & 1 & -1 \\ \hline
\end{array}
\equiv
\begin{array}{|c|c|c|c|}
\hline
1 & 1 & 0 & 0 \\ \hline
0 & 0 & 1 & 1 \\ \hline
1 & 0 & 0 & 0 \\ \hline
0 & 1 & 0 & 0 \\ \hline
0 & 0 & 1 & 0 \\ \hline
0 & 0 & 0 & 1 \\ \hline
1 & 0 & 0 & 0 \\ \hline
0 & 1 & 0 & 0 \\ \hline
0 & 0 & 1 & 0 \\ \hline
0 & 0 & 0 & 1 \\ \hline
\end{array}
>
\begin{array}{|c|c|c|c|}
\hline
1 & 1 & 0 & 0 \\ \hline
0 & 0 & 1 & 1 \\ \hline
\sqrt{2} & 0 & 0 & 0 \\ \hline
0 & \sqrt{2} & 0 & 0 \\ \hline
0 & 0 & \sqrt{2} & 0 \\ \hline
0 & 0 & 0 & \sqrt{2} \\ \hline
\end{array}
$$

Equivalent error for all queries

Lower error for all queries

Wavelet Y — $||Y||_1 = 3$

Y′ — $||Y'||_1 = 3$

Y″ — $||Y''||_1 = 2.414$

The haar wavelet observation matrix **Y** is **dominated** by alternative matrix **Y″**.

The matrix mechanism

Given a workload $\mathbf{W}$, and any full-rank strategy matrix $\mathbf{A}$, the following randomized algorithm is ε-differentially private:

$$\mathbf{Matrix_A(W,x) = Wx + (||A||_1 / \varepsilon)\, WA^+ b}$$

$\mathbf{b=Lap(1)}$

instantiated with measurements A

true answer

scaling by $||\mathbf{A}||_1$

transformation by $\mathbf{WA^+}$

Compare with the Laplace mechanism:

$$\mathbf{Laplace(W,x) = Wx + (||W||_1 / \varepsilon) b}$$

OPT_{MM}: Matrix mechanism optimization [Li et al., 2010]

- For any $\mathbf{A}$ that supports $\mathbf{W}$, expected **total squared error** is:

$$Error(\mathbf{W}, \mathbf{A}) = \underbrace{(2/\epsilon^2) \left\|\mathbf{A}\right\|_1^2}_{\text{Measurement error}} \underbrace{\left\|\mathbf{W}\mathbf{A}^+\right\|_F^2}_{\text{Reconstruction Error}}$$

Error independent of the input data

Matrix Mechanism optimization is hard

- To find the $\mathbf{A}$ that minimizes error on $\mathbf{W}$:

$$\underset{\mathbf{A}}{\text{minimize}} \quad \|\mathbf{A}\|_1^2 \left\|\mathbf{W}\mathbf{A}^+\right\|_F^2 \quad \longleftarrow \text{Expected Error}$$

$$\text{subject to} \quad \mathbf{W}\mathbf{A}^+\mathbf{A} = \mathbf{W} \quad \longleftarrow \mathbf{A} \text{ supports } \mathbf{W}$$

- **It is hard for a number of reasons:**

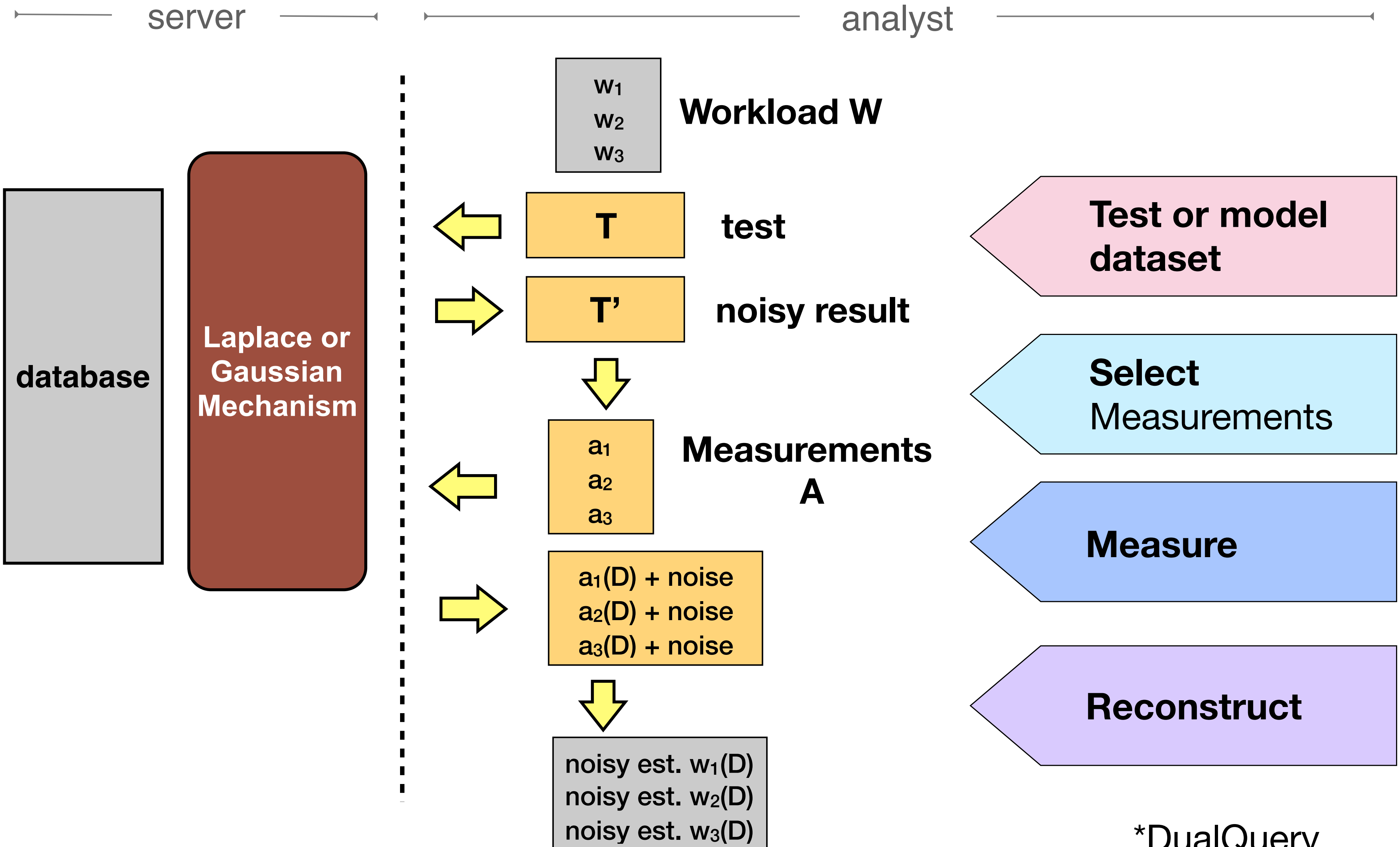
1. There are **many parameters** to optimize
2. The pseudo inverse is **expensive to compute** and **not well-behaved**
3. The constraints are **hard to encode**
4. The problem is **not smooth or convex**

Optimal selection of observations

Objective: given workload $\mathbf{W}$, find the observation matrix $\mathbf{A}$ that minimizes the **total** error.

Privacy	Optimization Objective	Problem Type	Runtime
ε DP	Given $\mathbf{W}$ consisting of data cube queries, choose $\mathbf{A}$ consisting of data cube queries to minimize simplified error measure. **[Ding, SIGMOD '11]**	set-cover approx	O(n)
ε DP	Given $\mathbf{W}$, choose $\mathbf{A}$ to minimize $\mathbf{TotalError_A(W)}$ **[Li, PODS '10]**	SDP w/ rank constraints	$O(n^8)$
(ε,δ) DP	Given $\mathbf{W}$, choose $\mathbf{A}$ to minimize $\mathbf{TotalError_A(W)}$ **[Li, PODS '10]**	SDP	$O(n^8)$
ε DP	Given $\mathbf{W}$, choose $\mathbf{AB}\approx\mathbf{W}$ to minimize $\mathbf{TotalError_A(AB)}$ **[Yuan, VLDB '12]**	bi-convex opt	$O(n^4)$
(ε,δ) DP	Given $\mathbf{W}$, choose optimal scaling of eigenvectors of $\mathbf{W}$ to minimize $\mathbf{TotalError_A(W)}$ **[Li, PVLDB '12]**	convex opt	$O(n^4)$

Approach 2: data-adaptive mechanisms



Selected data-adaptive mechanisms

Workload	Measurements	Citation
1D range queries	approx. v-optimal histogram	[Xu, ICDE ’12]
2D range queries	kd-tree queries	[Xiao, SDM ‘10]
2D range queries	hybrid kd-tree queries	[Cormode, ICDE ’12]
Marginals	scaled workload queries	[Xiao, SIGMOD ’11]
Linear queries	subset of workload	[Hardt, NIPS ’12]
Any (none specified)	stats of Bayes Net	[Zhang, SIGMOD ’14]
1D/2D range queries	tree queries; reduced domain	[Li, VLDB ’14]
Linear queries	minimum payoff records	[Gaboardi, ICML ’14]

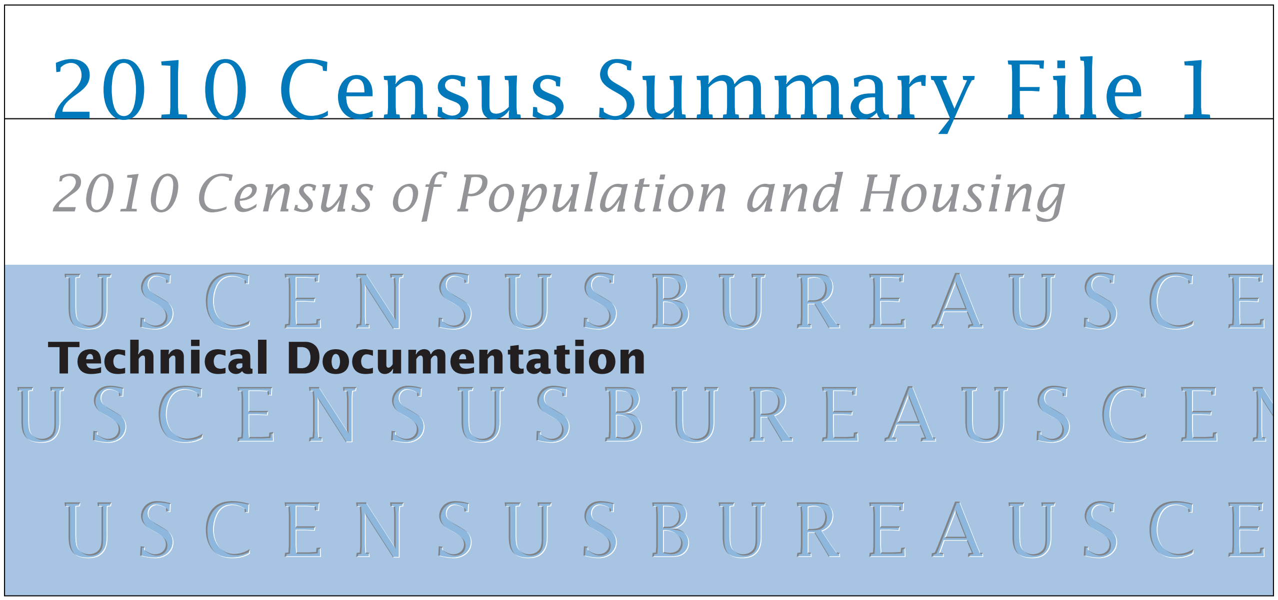
Comparison of approaches

Data-agnostic	Data-adaptive
Most fit the “select-measure-reconstruct” paradigm	Greater variety of techniques
Workload query error easily computable and non-sensitive.	Workload query error is data-dependent and sensitive.
Unbiased query answers	Reduce variance by introducing bias into query answers
Lower error in “high signal” settings	Lower error in “low signal” settings
Scalability challenges	Scalability challenges (with some exceptions)

Outline

1. Algorithm landscape
2. **Motivating challenge: a Census workload**
3. Scaling the matrix mechanism
4. Results on the Census workload
5. Data-adaptive algorithms and trade-offs
6. Open problems

Census of Population and Housing


2010 Census Summary File 1

2010 Census of Population and Housing

Technical Documentation

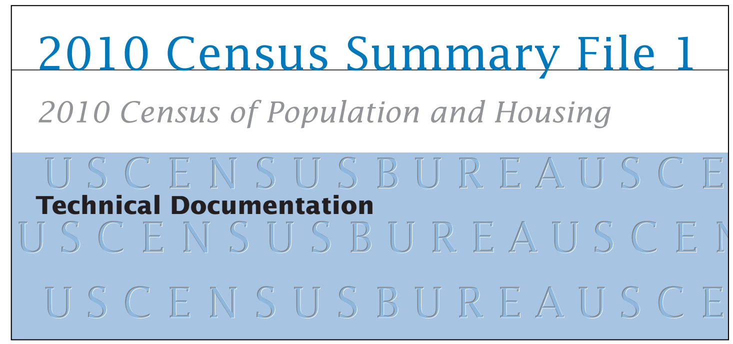
Describes **Persons** and their **Households**

SF1 = “Summary File 1”

Example data and workload

- **Persons** table:
  - sex (2)
  - relation (17)
  - age (115)
  - race/ethnicity (126)
  - geography-state (52)
  - geography-tract (73,768)
  - geography-blocks (10,620,683)

Workload


2010 Census Summary File 1

2010 Census of Population and Housing

Technical Documentation

4151 predicate counting queries on **Persons**

Person table, in vector form

- **Persons** table:
  - sex (n_1=2)
  - relation (n_2=17)
  - age (n_3=115)
  - race/ethnicity (n_4=126)
  - geography-state (n_5=52)
  - geography-tract (n_6=73,768)
  - geography-blocks (n_7=10,620,683)

	Num. entries in data vector "domain size"
national (sex, relation, age, race/ethnicity)	492,660
+ geography-state	25,618,320
+ geography-tract	36,342,542,880
+ geography-blocks	5,232,385,686,780

Product workloads

Given a set of predicates on each attribute, a **product workload** consists of all predicate queries that conjunctively combine one predicate on each attribute.

W_{age}

```
age>18
age=65
age∈[18..25]
```

X

W_{race}

```
race∈[1,2,3]
race=3
```

=

```
age>18 AND race∈[1,2,3]
age>18 AND race=3
age=65 AND race∈[1,2,3]
age=65 AND race=3
age∈[18..25] AND race∈[1,2,3]
age∈[18..25] AND race=3
```

Note: marginals are product workloads where predicate sets are either {True} or “Identity”:

$$I_{age} \;\; \text{x} \;\; I_{race} \;\; \text{x} \;\; \{\text{True}\}_{relp} \;\; \text{x} \;\; \{\text{True}\}_{sex}$$

Product workload example

- Many SF1 “tables” can be represented as product workloads

- For example, table P12 (excluding the Total) is:

P12. **SEX BY AGE [49]**
Universe: Total population
Total:
Male:
Under 5 years
5 to 9 years
10 to 14 years
15 to 17 years
18 and 19 years
20 years

$$W_{sex} \times W_{age} \times W_{racethn} \times W_{relp}$$

W_{sex}	W_{age}	$W_{racethn}$	W_{relp}
sex=M sex=F	True age∈[0,5) age∈[5,9) … age∈[85,115]	True	True

Products and Union of Products

- A **product workload** can encode a **cartesian product of counting queries** in which conditions are combined conjunctively. Examples include:
  - All multi-dimensional range queries
  - a single marginal
  - all marginals
- A **union of products workload** can encode an **arbitrary collection of counting queries** in which conditions are combined conjunctively. Examples include:
  - Arbitrary collection of multi-dimensional range queries
  - Arbitrary collection of marginals
  - Census Summary File 1 (SF1): union of 32 product workloads, sensitivity=50

Census SF1 workload (Person queries)

sex	age	race	ethnicity	relp	geo
I	{coarse ranges}	T	T	T	{Block}
I	{u[illegible]	T	T	T	{Block}
T	[illegible]	[illegible]	[illegible]	[illegible]	{Block}
T	T	{race-comb}	T	T	{Block}
T	{over 18}	{race-comb}	I	T	{Block}
I	I	I	I	T	{Tract}
....					

Public Law 94-171: Important Redistricting data

Can we scale the matrix mechanism?

$\mathbf{x}$	$\leftarrow$	vectorize(R)	Input
$\mathbf{W}$	$\leftarrow$	vectorize(W)	
$\mathbf{A}$	$\leftarrow$	$\mathrm{OPT}_{\mathrm{MM}}(\mathbf{W})$	Select
Δ_A	$\leftarrow$	$\lVert\mathbf{A}\rVert_1$	Measure
$\mathbf{a}$	$\leftarrow$	$\mathbf{Ax}$	
$\mathbf{y}$	$\leftarrow$	$\mathbf{a} + \mathrm{Lap}(\Delta_A/\epsilon)$	
$\bar{\mathbf{x}}$	$\leftarrow$	$\mathbf{A}^{+}\mathbf{y}$	Reconstruct
ans	$\leftarrow$	$\mathbf{W}\bar{\mathbf{x}}$	

Can we scale the matrix mechanism?

Matrix Mechanism (MM)

$\mathbf{x}$	$\leftarrow$	vectorize(R)
$\mathbf{W}$	$\leftarrow$	vectorize(W)
$\mathbf{A}$	$\leftarrow$	$\mathrm{OPT}_{\mathrm{MM}}(\mathbf{W})$
Δ_A	$\leftarrow$	$\lVert\mathbf{A}\rVert_1$
$\mathbf{a}$	$\leftarrow$	$\mathbf{Ax}$
$\mathbf{y}$	$\leftarrow$	$\mathbf{a} + \mathrm{Lap}(\Delta_A/\epsilon)$
$\bar{\mathbf{x}}$	$\leftarrow$	$\mathbf{A}^{+}\mathbf{y}$
ans	$\leftarrow$	$\mathbf{W}\bar{\mathbf{x}}$

data vector is big

SF1-national	$\sim 10^6$
SF1-state	$\sim 10^7$

workload matrix is enormous

SF1-national	8 GB
SF1-state	22 TB

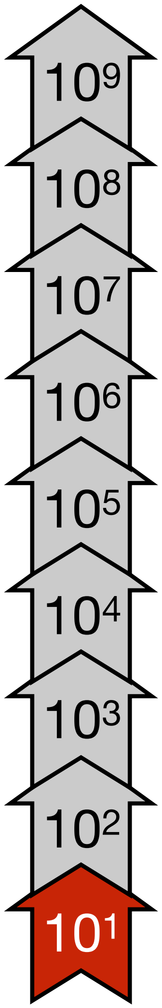
solve optimization problem?

also watch out for reconstruction

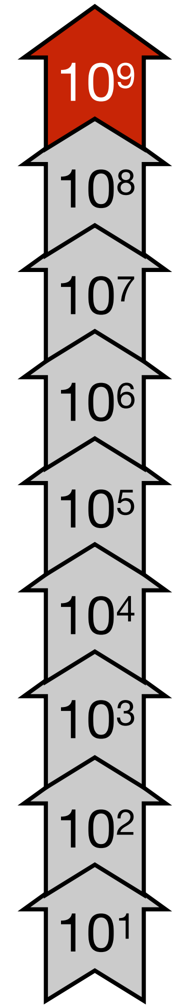
Outline

1. Algorithm landscape
2. Motivating challenge: a Census workload
3. **Scaling the matrix mechanism**
4. Results on the Census workload
5. Data-adaptive algorithms and trade-offs
6. Open problems

Matrix Mechanism vs. HDMM



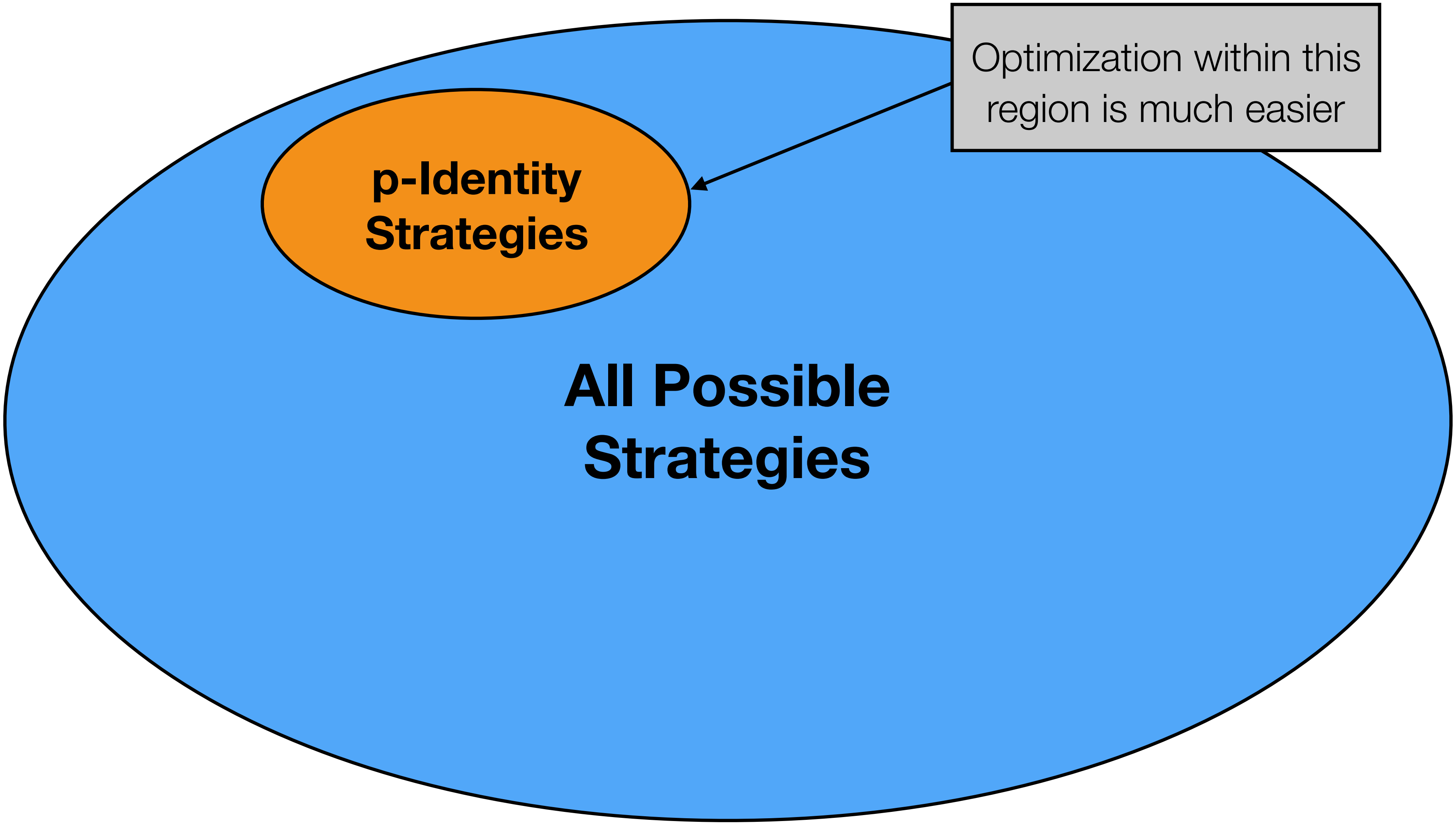
MM				HDMM		
$\mathbf{x}$	←	vectorize(R)	Input	$\mathbf{x}$	←	vectorize(R)
$\mathbf{W}$	←	vectorize(W)		$\mathbb{W}$	←	impl-vect(W)
$\mathbf{A}$	←	$\text{OPT}_{\text{MM}}(\mathbf{W})$	Select	$\mathbb{A}$	←	$\text{OPT}_{\text{HDMM}}(\mathbb{W})$
Δ_A	←	$\lVert\mathbf{A}\rVert_1$		Δ_A	←	$\text{sensitivity}(\mathbb{A})$
$\mathbf{a}$	←	$\mathbf{Ax}$	Measure	$\mathbf{a}$	←	$\text{mult}(\mathbb{A},\mathbf{x})$
$\mathbf{y}$	←	$\mathbf{a} + \text{Lap}(\Delta_A/\epsilon)$		$\mathbf{y}$	←	$\mathbf{a} + \text{Lap}(\Delta_A/\epsilon)$
$\bar{\mathbf{x}}$	←	$\mathbf{A}^{+}\mathbf{y}$	Reconstruct	$\bar{\mathbf{x}}$	←	$\text{LstSqr}(\mathbb{A}, \mathbf{y})$
ans	←	$\mathbf{W}\bar{\mathbf{x}}$		ans	←	$\text{mult}(\mathbb{W}, \bar{\mathbf{x}})$



Main obstacles ➠ solution:

1. OPT_{MM} is intractable ➠ local, parameterized search

OPT$_0$: Optimizing over p-Identity strategies

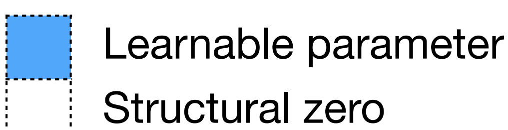


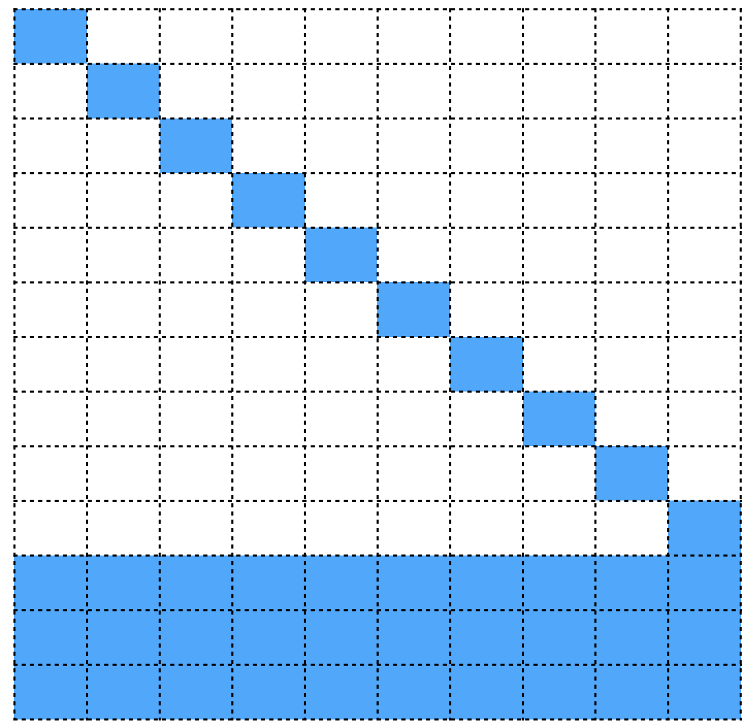
OPT$_0$: Optimizing over p-Identity strategies

- Key Idea: Instead of optimizing over all strategies, optimize over the space of "p-Identity" strategies:

$$\mathbf{A}(\boldsymbol{\Theta}) = \begin{bmatrix} \mathbf{I} \\ \boldsymbol{\Theta} \end{bmatrix} diag(1 + \mathbf{1}^T \boldsymbol{\Theta})^{-1}$$

Carefully designed to make optimization easier





OPT_0: Optimizing over p-Identity strategies

- Sensitivity is always 1 by construction:

$$||\mathbf{A}(\boldsymbol{\Theta})||_1 = 1$$

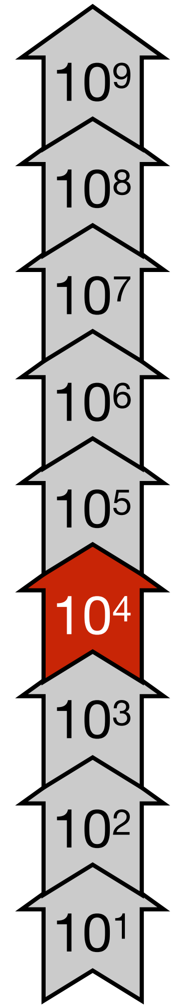
- **A** supports all workloads because it has full column rank:

$$\mathbf{W}\mathbf{A}^+\mathbf{A} = \mathbf{W} \quad \text{for all } \mathbf{A}(\boldsymbol{\Theta})$$

- Optimization is much simpler over this space:

$$\underset{\boldsymbol{\Theta}}{\text{minimize}} \left\| \mathbf{W}\mathbf{A}(\boldsymbol{\Theta})^+ \right\|_F^2$$

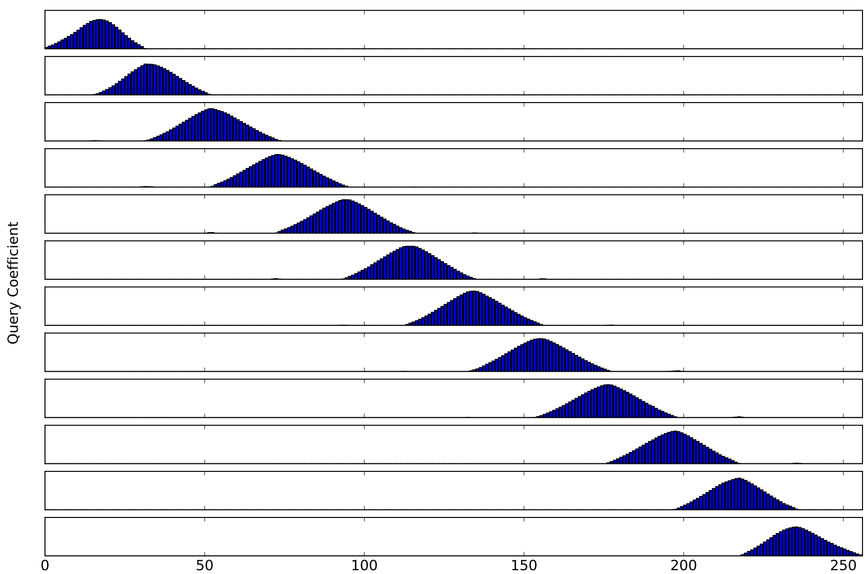
- Objective can be evaluated 240X faster by exploiting structure of **A(Θ)** (for n=8192, p=512)



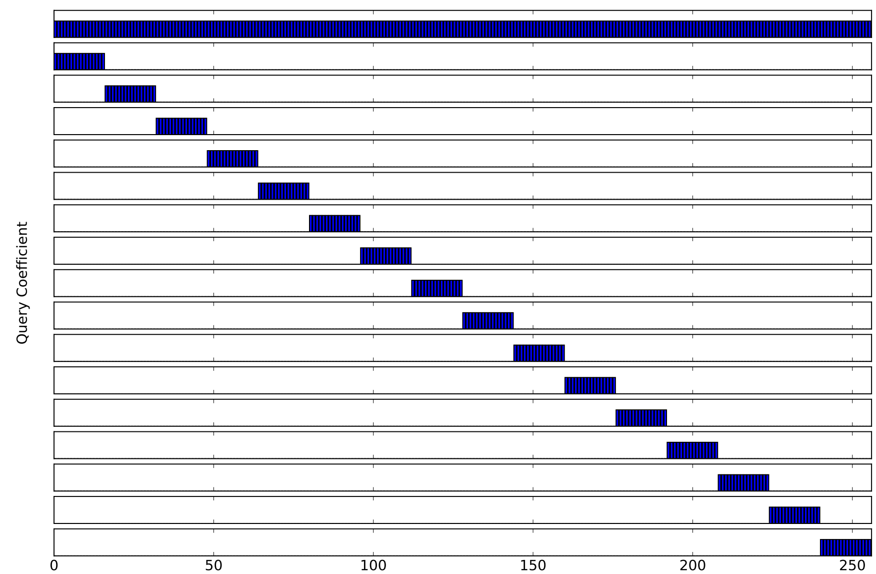
Visualizing OPT_0 output

Workload of all range queries on 1D domain n=256

The strategy computed by OPT_0 for this workload (p=12)



A competing strategy, H_{16}, using hierarchical queries with 16-way branching



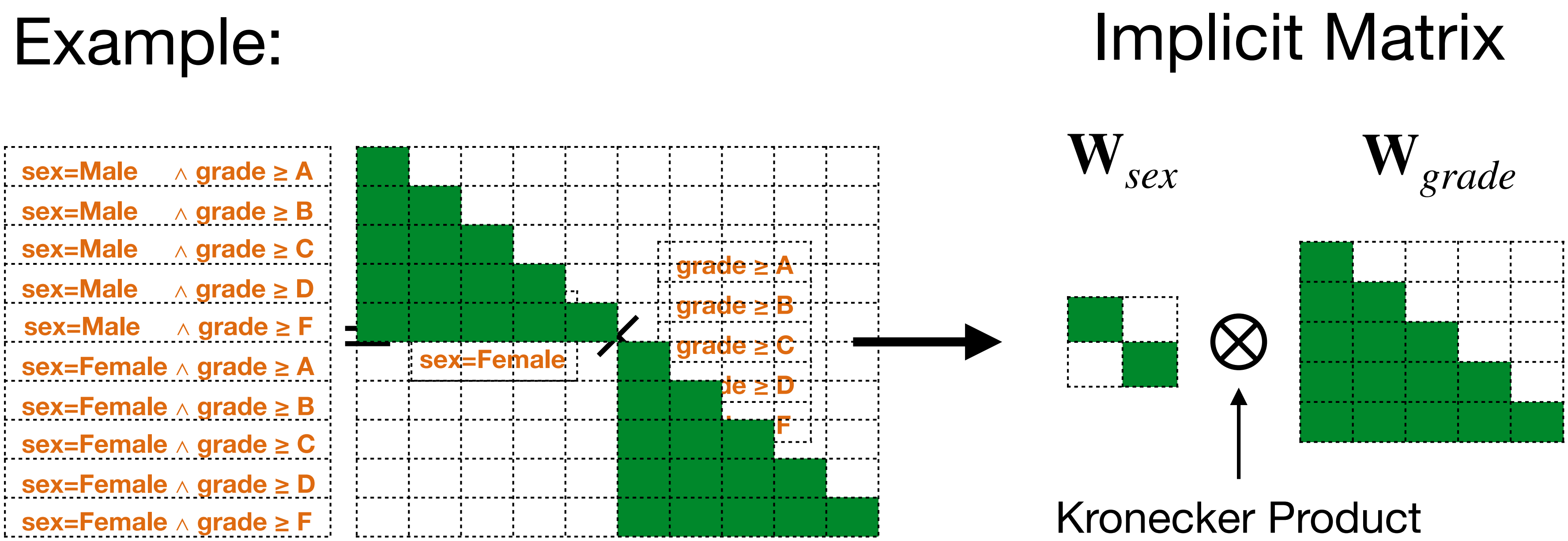
Both strategies include the 256 identity queries (not shown)

Error on Prefix workload

Domain	HDMM	Identity	H2	Privelet	HB	GreedyH
128	1.00	1.80	1.79	1.78	1.80	1.20
256	1.00	2.18	1.79	1.78	1.22	1.24
512	1.00	2.68	1.80	1.79	1.28	1.41
1024	1.00	3.34	1.80	1.80	1.34	1.49
2048	1.00	4.18	1.80	1.79	1.42	1.71
4096	1.00	5.25	1.78	1.78	1.22	1.84
8192	1.00	6.40	1.71	1.70	1.20	2.09

Implicit workload representation

- Idea: we can store some workloads more efficiently



We can **represent large multi-dimensional workloads** by storing only small sub-workloads

Implicit representations are extremely compact

Workload	Explicit size	Implicit size
P12 table	96 MB	24 KB
SF1-national	8 GB	335 KB
SF1-state	22 TB	687 KB

Properties of Kronecker products

$$(\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C})$$ Associativity

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$$ Matrix multiplication

$$(\mathbf{A} \otimes \mathbf{B})^{+} = \mathbf{A}^{+} \otimes \mathbf{B}^{+}$$ Pseudo inverse

$$||\mathbf{A} \otimes \mathbf{B}|| = ||\mathbf{A}|| \cdot ||\mathbf{B}||$$ Matrix norm

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} \qquad \sigma_{ij}^{\mathbf{C}} = \sigma_i^{\mathbf{A}} \sigma_j^{\mathbf{B}}$$ Singular values

OPT$_\otimes$: Optimizing Kronecker product workloads

- Given a Kronecker product workload:

$$\mathbb{W} = \mathbf{W}_1 \otimes \ldots \otimes \mathbf{W}_d$$

- What can we do?
  - Finding a p-Identity strategy won't work - workload may be too large to represent as a dense matrix

- A natural idea: try to find a Kronecker product strategy

$$\mathbb{A} = \mathbf{A}_1 \otimes \ldots \otimes \mathbf{A}_d$$

OPT$_{\otimes}$: Optimizing over Kronecker product strategies

- Given a Kronecker product workload and strategy:

$$\mathbb{W} = \mathbf{W}_1 \otimes \ldots \otimes \mathbf{W}_d \qquad \mathbb{A} = \mathbf{A}_1 \otimes \ldots \otimes \mathbf{A}_d$$

- Sensitivity and error profile decompose over the factors:
- Expected error decomposes over the factors

$$||\mathbb{A}||_1 = \prod_{i=1}^{d} ||\mathbf{A}_i||_1$$

$$Error(\mathbb{W}, \mathbb{A}) = \prod_{i=1}^{d} Error(\mathbf{W}_i, \mathbf{A}_i)$$

- SVD lower bound decomposes over the factors:

$$||\mathbb{W}\mathbb{A}^{+}||_F = \prod_{i=1}^{d} ||\mathbf{W}_i \mathbf{A}_i^{+}||_F$$

$$SVDB(\mathbb{W}) = \prod_{I=1}^{d} SVDB(\mathbf{W}_i)$$

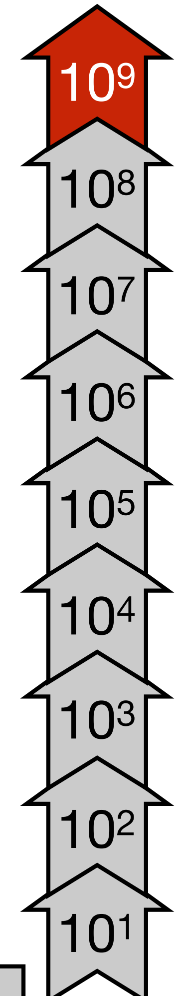
OPT$_\otimes$: Optimizing over Kronecker product strategies

- Given a Kronecker product workload and strategy:

$$\mathbb{W} = \mathbf{W}_1 \otimes \ldots \otimes \mathbf{W}_d \qquad\qquad \mathbb{A} = \mathbf{A}_1 \otimes \ldots \otimes \mathbf{A}_d$$

- Expected error decomposes over the factors

$$Error(\mathbb{W}, \mathbb{A}) = \prod_{i=1}^{d} Error(\mathbf{W}_i, \mathbf{A}_i)$$



To minimize error:

solve d small optimization problems over the sub-workloads
(which we can do efficiently using p-Identity strategies)

$\text{OPT}_\otimes$: Optimizing over Kronecker product strategies

- Given a union of Kronecker product workload:

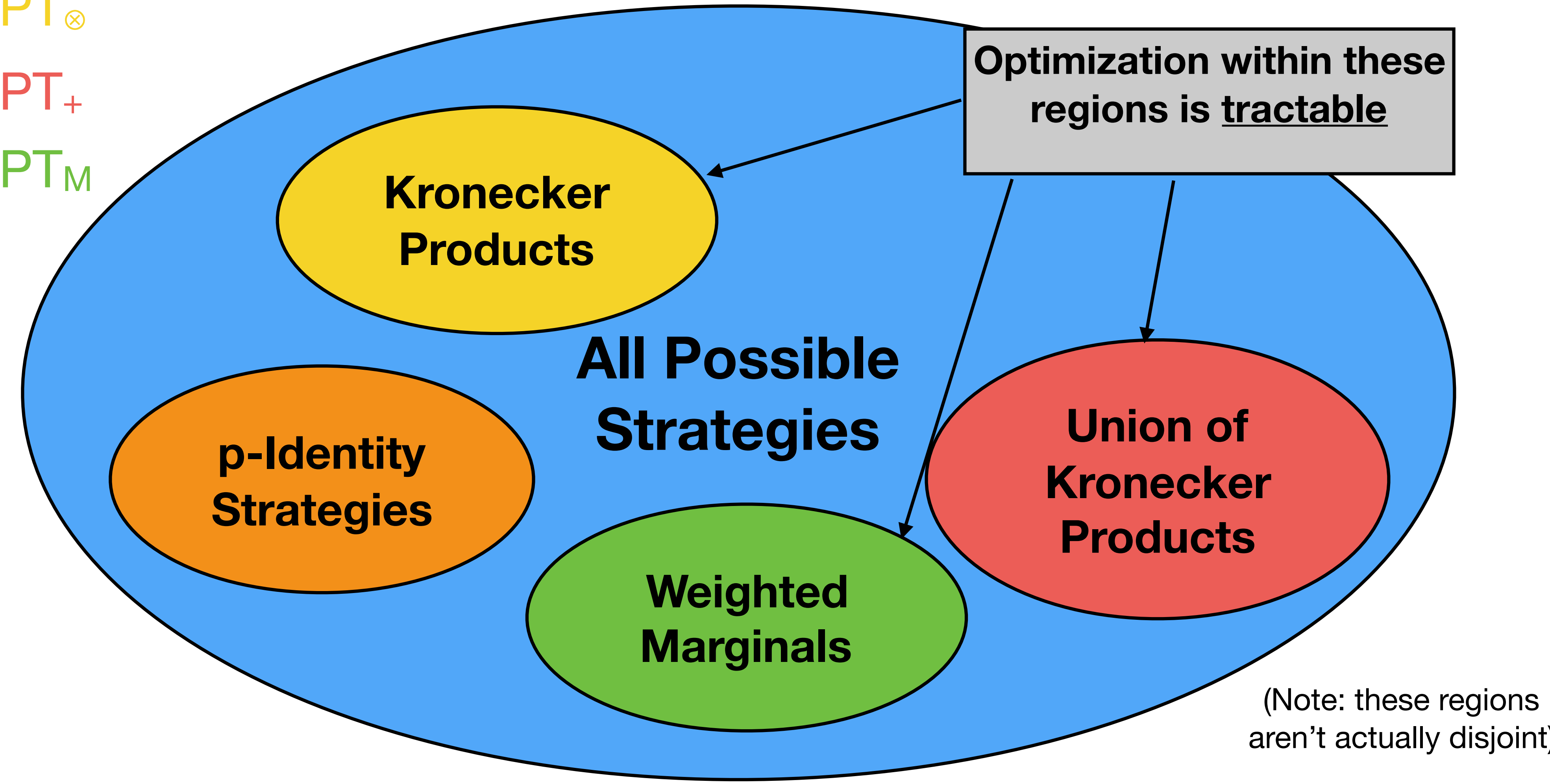
$$\mathbb{W} = \begin{bmatrix} \mathbb{W}^{(1)} \\ \vdots \\ \mathbb{W}^{(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1^{(1)} \otimes \dots \otimes \mathbf{W}_d^{(1)} \\ \vdots \otimes \dots \otimes \vdots \\ \mathbf{W}_1^{(k)} \otimes \dots \otimes \mathbf{W}_d^{(k)} \end{bmatrix}$$

- There are **three strategy optimization routines**:
  1. $\mathbf{OPT_+}$ - searches over union of Kron product of p-Identity strategies
  2. $\mathbf{OPT_\otimes}$ - searches over Kron product of p-Identity strategies
  3. $\mathbf{OPT_M}$ - searches over weighted marginals strategies

$\}$ Makes calls to OPT_0

Optimizing Union of Product Workloads

- $OPT_{\otimes}$
- OPT_{+}
- OPT_{M}



Do these regions contain high quality strategies?

It depends on the workload, but experimental evidence suggests **Yes**.

OPT$_+$: Optimizing union of Kronecker product strategies

- Simple idea: optimize each sub workload separately:

$$\mathbb{A}^{(j)} = OPT_{\otimes}(\mathbb{W}^{(j)})$$

- And form a union of Kronecker strategy:

$$\mathbb{A} = \begin{bmatrix} \mathbb{A}^{(1)} \\ \vdots \\ \mathbb{A}^{(k)} \end{bmatrix}$$

$$Error(\mathbb{W}, \mathbb{A}) \leq \sum_{j} Error(\mathbb{W}^{(j)}, \mathbb{A}^{(j)})$$

OPT$_\otimes$: Optimizing over Kronecker product strategies

- Given a Kronecker product strategy:

$$\mathbb{A} = \mathbf{A}_1 \otimes \ldots \otimes \mathbf{A}_d$$

- Expected error still decomposes for a union of Kronecker workload:

$$\begin{aligned} Error(\mathbb{W}, \mathbb{A}) &= \sum_{j=1}^{k} Error(\mathbb{W}^{(j)}, \mathbb{A}) \\ &= \sum_{j=1}^{k} \prod_{i=1}^{d} Error(\mathbf{W}_i^{(j)}, \mathbf{A}_i) \end{aligned}$$

- Thus we can solve the optimization problem efficiently

OPT$_{\mathbf{M}}$: Optimizing marginals strategies

- Marginals are Kronecker products:

$$\mathbb{M}_{1100} = \mathbf{I} \otimes \mathbf{I} \otimes \mathbf{T} \otimes \mathbf{T}$$

- A collection of weighted marginals is a union of Kronecker products:

$$\mathbb{M}(\theta) = \begin{bmatrix} \theta_1(\mathbf{T} \otimes \cdots \otimes \mathbf{T}) \\ \vdots \\ \theta_{2^d}(\mathbf{I} \otimes \cdots \otimes \mathbf{I}) \end{bmatrix}$$

$$Error(\mathbb{W}, \mathbb{M}(\theta)) = ||M(\theta)||_1^2 ||\mathbb{W}\mathbb{M}(\theta)^+||_F^2$$

$$\Big(\sum_i \theta_i\Big)^2$$

Can compute pseudo inverse efficiently by exploiting structure

Overview: running HDMM

Given: schema of R, and (logical) workload W

1. **Represent workload implicitly as union of Kronecker products**
   - Combine columns if necessary
2. **Select best strategy from $\text{OPT}_{\otimes}$, OPT_{+}, and OPT_{M}**
   - (Optional) perform multiple random restarts
3. **Run the matrix mechanism:**
   - Measure queries in $\mathbb{A}$ with Laplace mechanism
   - Reconstruct W answers (by solving least squares prob

All 3D range queries → $\text{OPT}_{\otimes}$

All up-to-3 way marginals → OPT_{M}

Some other workloads → OPT+

How close to optimal are we?

- For (ε, δ)-differential privacy:
  - We have algorithms that can find globally optimal strategy
  - For all 2D range queries, we can get within a factor 1.04 of the SVD bound with a Kronecker product strategy.
- For ε-differential privacy:
  - Algorithms are approximate
  - 2-3X difference between lower bounds and what we can currently achieve
  - Open problem: need better bounds and/or optimization routines to close gap in (ε, 0)-differential privacy

Outline

1. Algorithm landscape
2. Motivating challenge: a Census workload
3. Scaling the matrix mechanism
4. **Results on the Census workload**
5. Data-adaptive algorithms and trade-offs
6. Open problems

More accuracy results: multi-dimensional workloads

- HDMM is one of the only algorithms that is general and scalable enough to handle complex multi-dimensional workloads
- HDMM offers lower error than competing methods

Dataset/ Domain	Workload	HDMM Error	Best competitor	
			Error	Method
CPH 2 x 2 x 17 x 51 x 63 x 115	**SF1**	1.00	3.07	Identity
	SF1+	1.00	3.15	Identity
Adult 2 x 5 x 16 x 20 x 75	**All Marginals**	1.00	1.38	Identity
	2-way Marginals	1.00	2.01	DataCube
CPS 2 x 4 x 7 x 50 x 100	**All Range Marginals**	1.00	1.49	Identity
	2-way Range Marginals	1.00	5.79	Identity

Many additional Census challenges

- Materializing data vector is prohibitive for full geography.
- Sophisticated post-processing is required on HDMM output: non-negativity, consistency (structural zeros and other known counts).
- Workload “tuning”:
  - What if we want lower error for sub-workload X?
  - What if we omit sub-workload Y? Is error improved elsewhere?
- Multiple releases: optimize and release sub-workload X; later, optimize and release related sub-workload Y consistent with X.
- Error rates can be computed and published, but how should they be communicated and utilized by stakeholders?

Tuning workload error

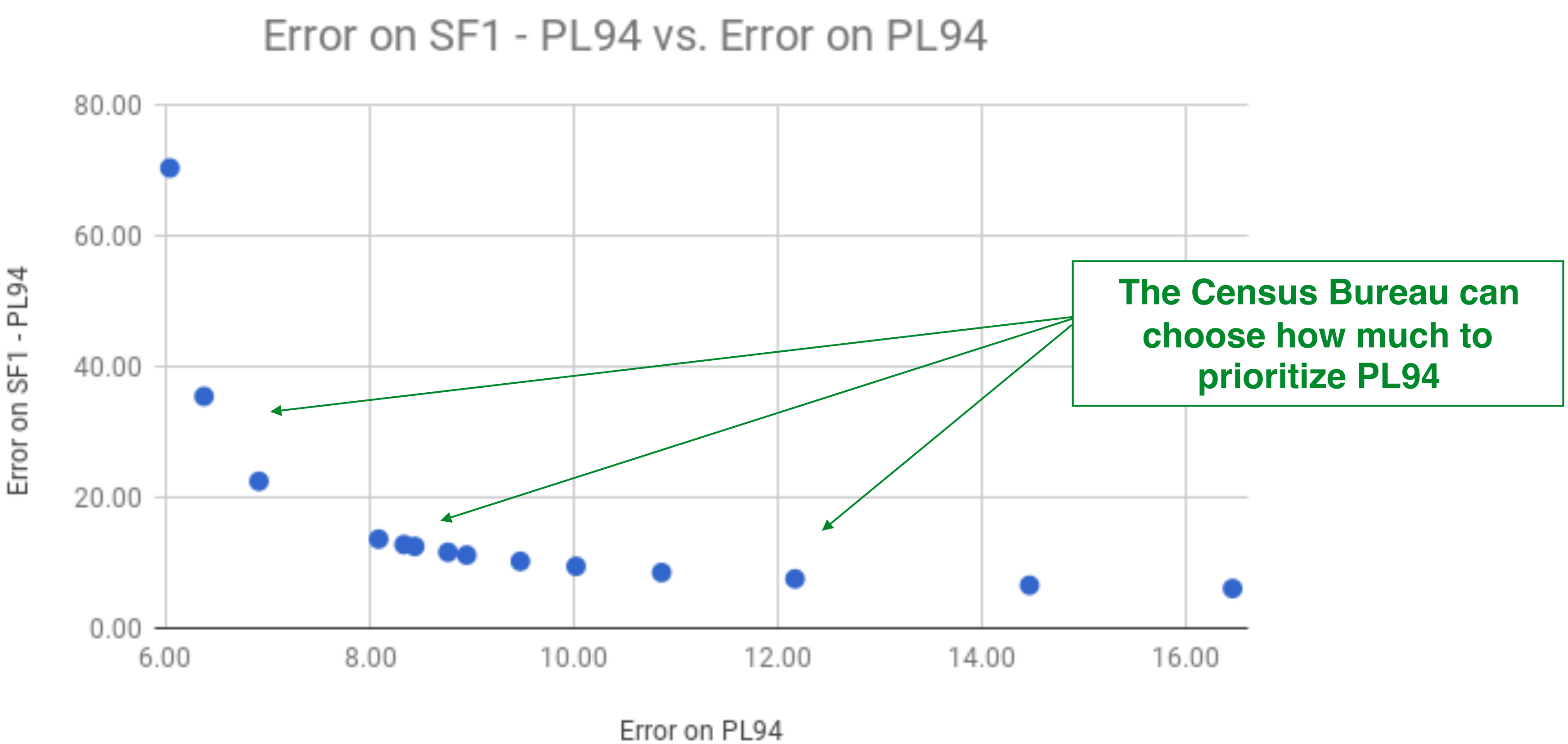
- The PL94 queries are an important subset of the SF1 workload.
  - PL94: 288 queries
  - SF1: 4151 queries

Optimized Workload	Avg. Per Query Error On …	
SF1	SF1	7.28
	PL94	16.45
	SF1 - PL94	6.07
PL94	PL94	3.91

$\varepsilon = 1$

Tuning workload error

- Optimizing for a workload in which PL94 is weighted
  - W = c*PL94 + 1*SF1 for positive constant c



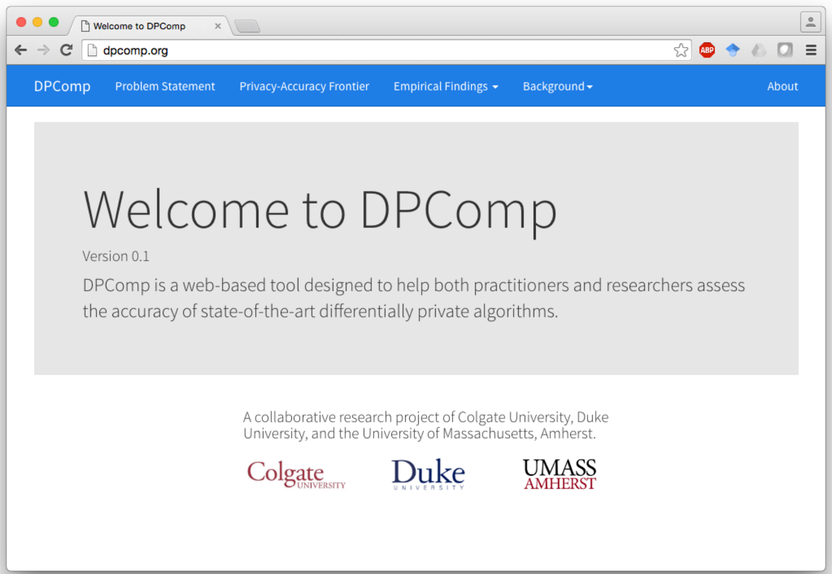
Outline

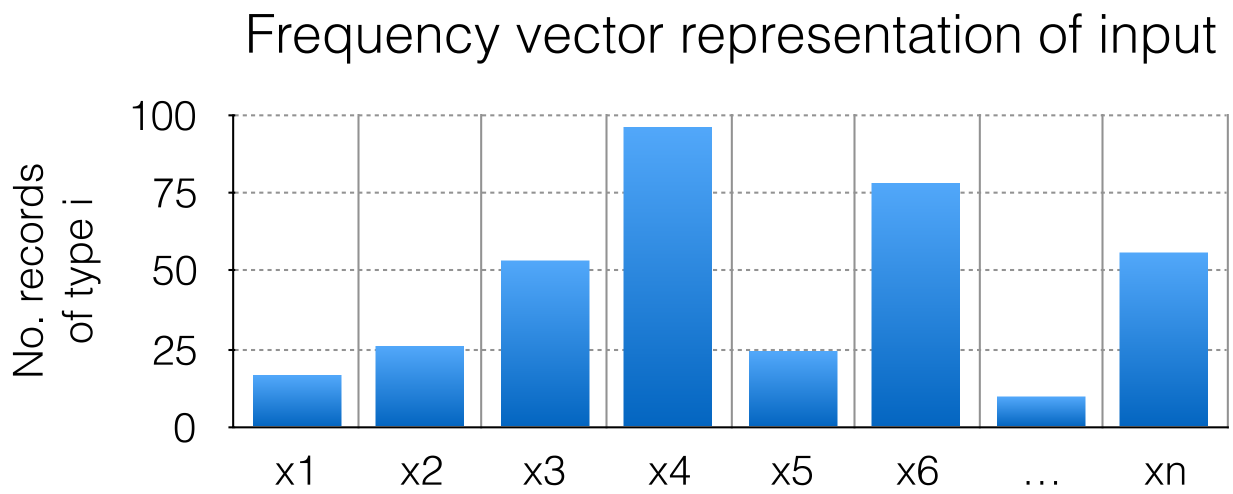
1. Algorithm landscape
2. Motivating challenge: a Census workload
3. Scaling the matrix mechanism
4. Results on the Census workload
5. **Data-adaptive algorithms and trade-offs**
6. Open problems

Data-adaptive mechanisms

- Understanding and evaluating data-adaptive algorithms is complex.

- The differential privacy community lacks benchmarks and standards for empirical evaluation.





Properties:

- **domain size**: length of frequency vector
- **scale**: total number of records in database
- **shape**: the frequency vector normalized by scale.

Desideratum: datasets that are diverse with respect to all three properties.

Data-dependent algorithms for low-dimensional linear queries

Uniform	**baseline**	Noisy total count; uniformity
MWEM	**[Hardt ’12]**	Multiplicative Weights Exp. Mech.
AHP	**[Zhang ’14]**	Private data reduction; measurement
DAWA	**[Li ’14]**	Private data reduction; measurement
PHP	**[Acs ’12]**	Private data reduction; measurement
QuadTree	**[Cormode ’12]**	2D adaptive grid-based techniques
UGrid	**[Qardaji ’13]**	2D adaptive grid-based techniques
AGrid	**[Qardaji ’13]**	2D adaptive grid-based techniques
EFPA	**[Acs ’12]**	Fourier; top-k coefficients

Error metric

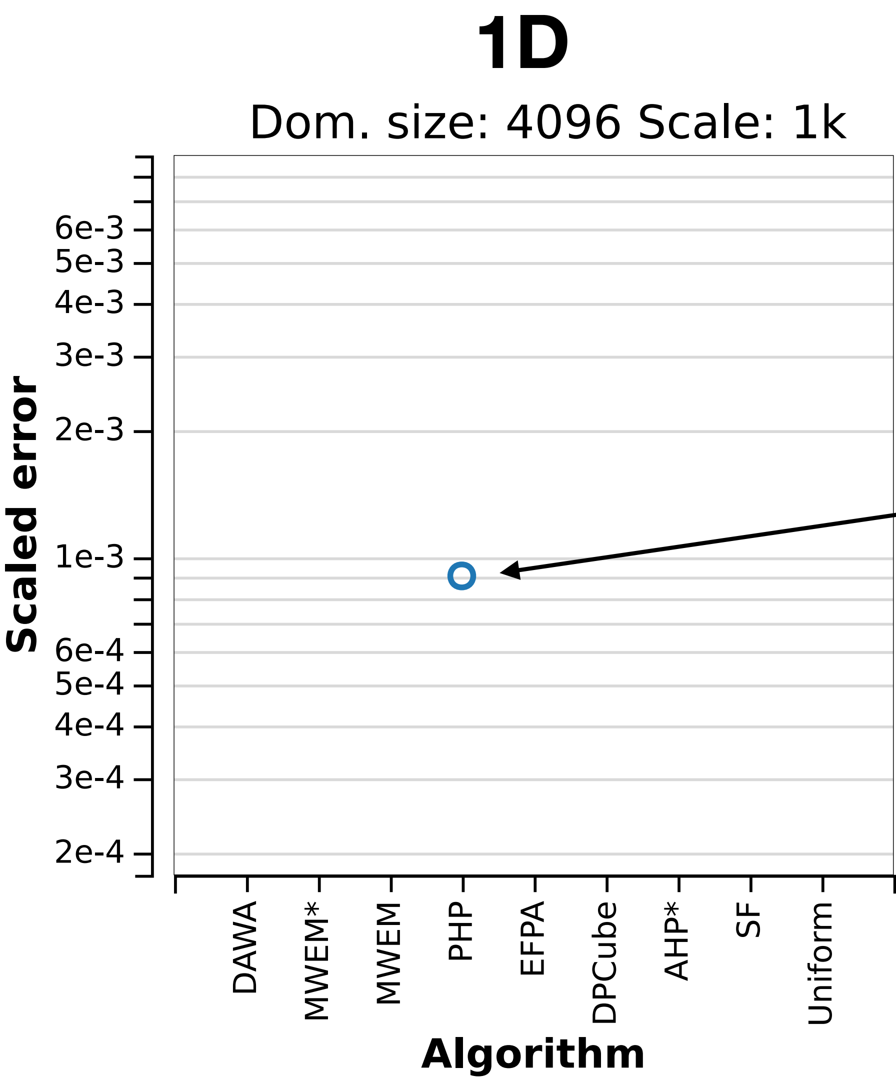
Definition 7 (Scaled Average Per-Query Error). *Let* $\mathbf{W}$ *be a workload of* q *queries,* $\mathbf{x}$ *a data vector and* $s = \|\mathbf{x}\|_1$ *its scale. Let* $\hat{\mathbf{y}} = \mathcal{K}(\mathbf{x}, \mathbf{W}, \epsilon)$ *denote the noisy output of algorithm* $\mathcal{K}$*. Given a loss function* L*, we define* scale average per-query error *as* $\frac{1}{s \cdot q} L(\hat{\mathbf{y}}, \mathbf{W}\mathbf{x})$.

Example (scaled error):

	Scale	Absolute Error	Scaled Absolute Error
Dataset 1	1,000	100	0.100
Dataset 2	100,000	100	0.001

Scaled error is also error in units of a "population percentage"

Variation with “shape”



Error for a dataset

Workload: Prefix

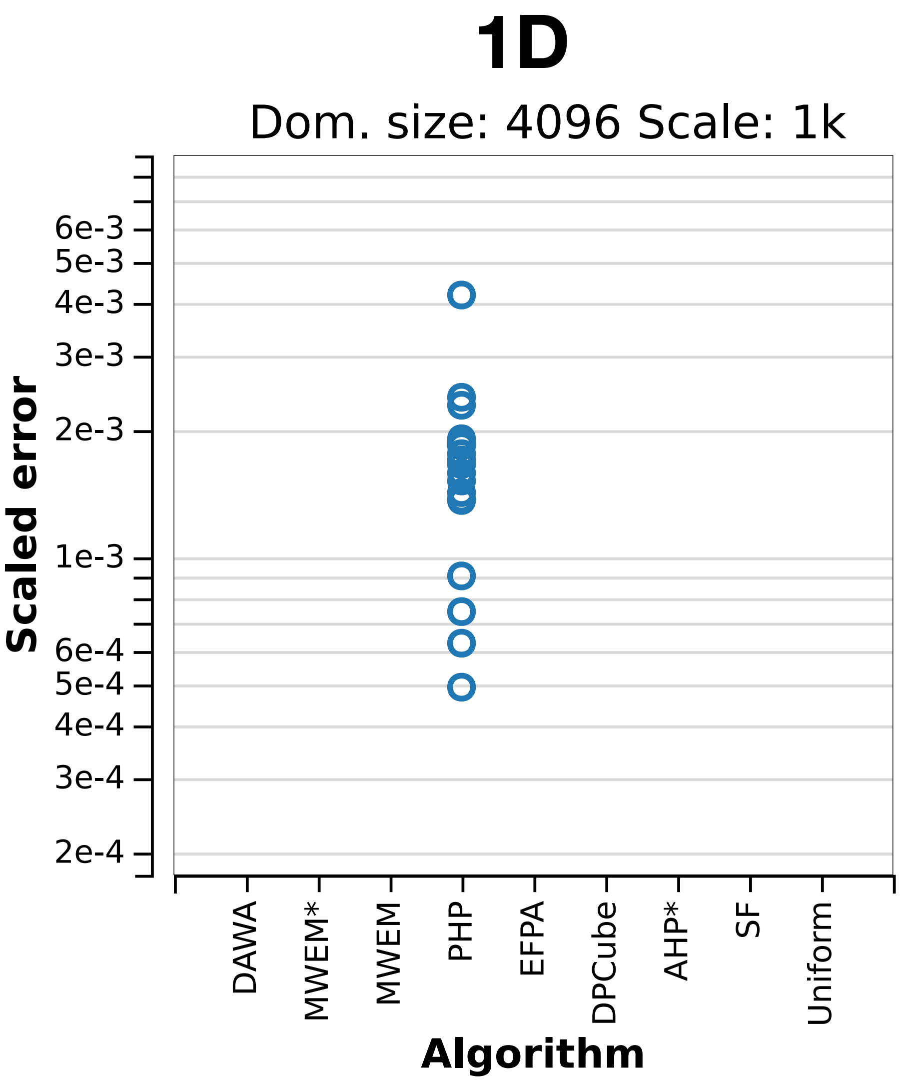
Shape: Patent

Domain size: 4096

Scale: 1000

(ε=0.1 throughout)

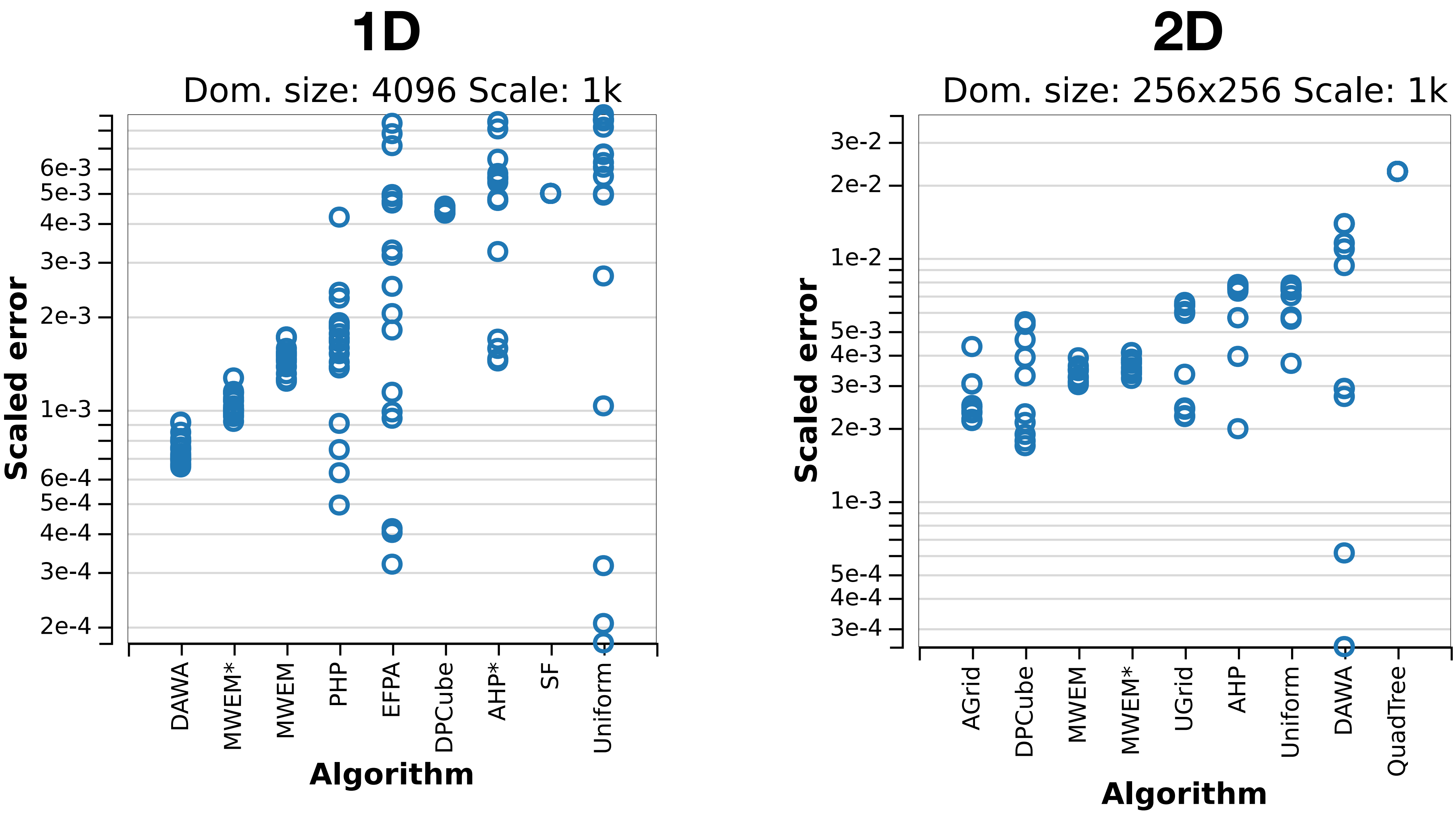
Variation with shape



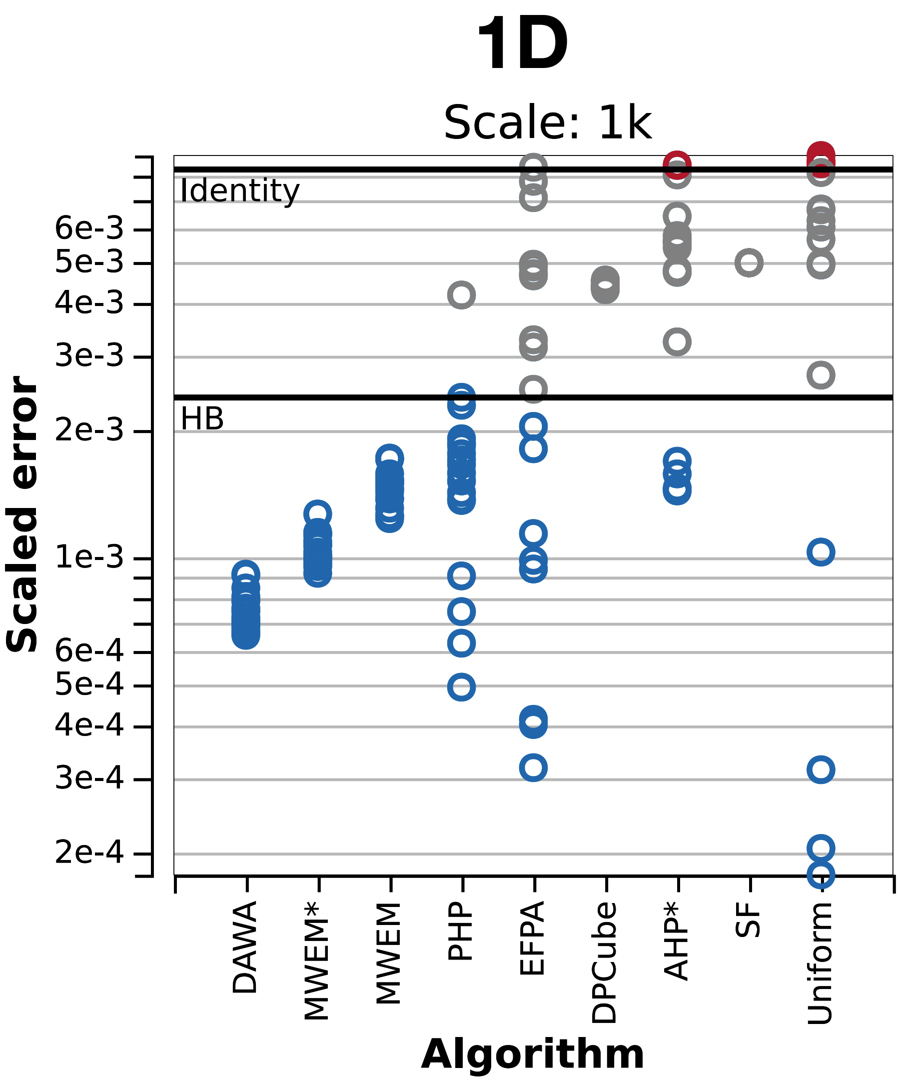
Variation across shape

(for fixed dimension, domain size, scale)

Algorithm error varies significantly with dataset shape



Data-independent alternatives



Data independent yardsticks

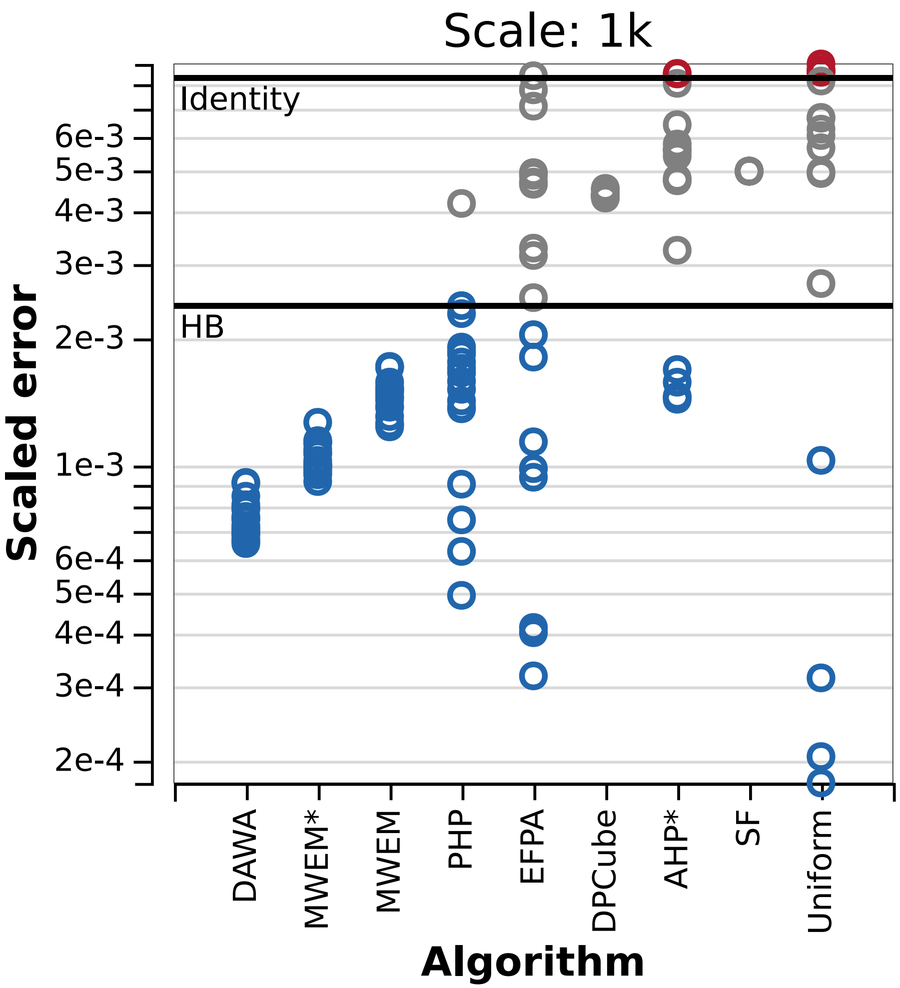
Identity: Laplace noise added to frequency vector **x**

HB: hierarchy of noisy counts

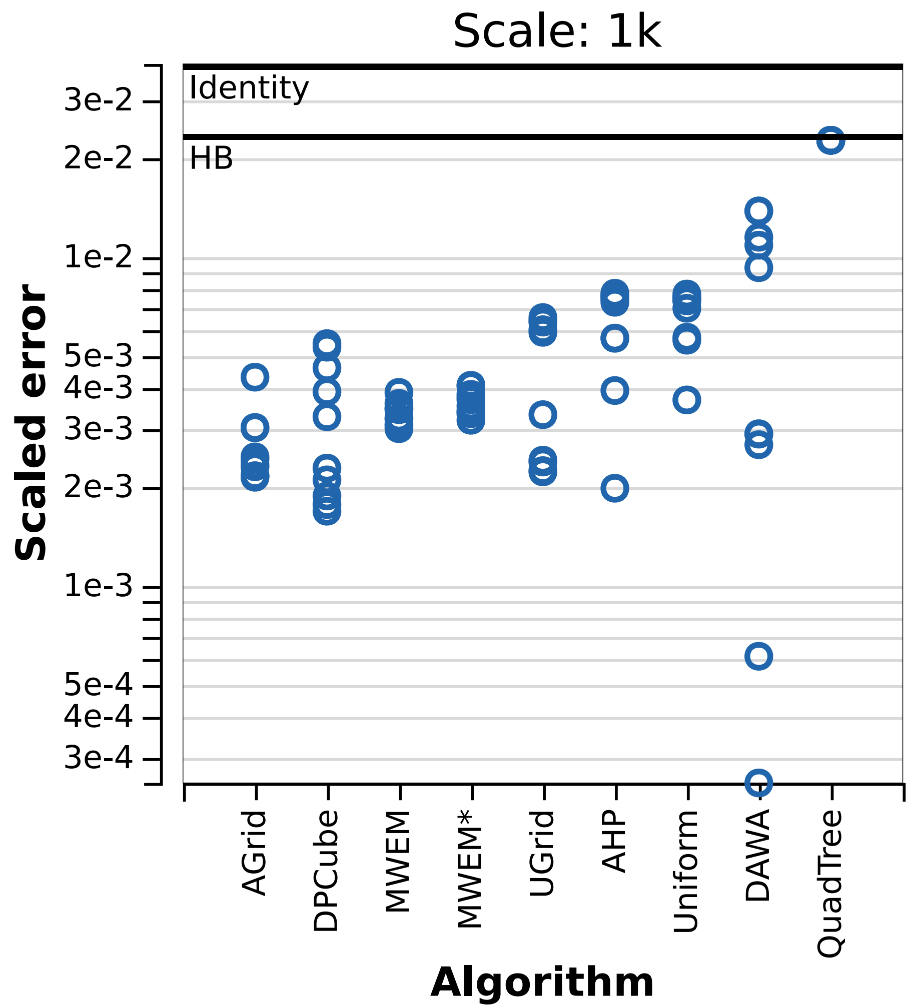
[Qardaji et al. ICDE 2013]

Data-dependence can offer significant improvements in error (at smaller scales or lower epsilon).

1D

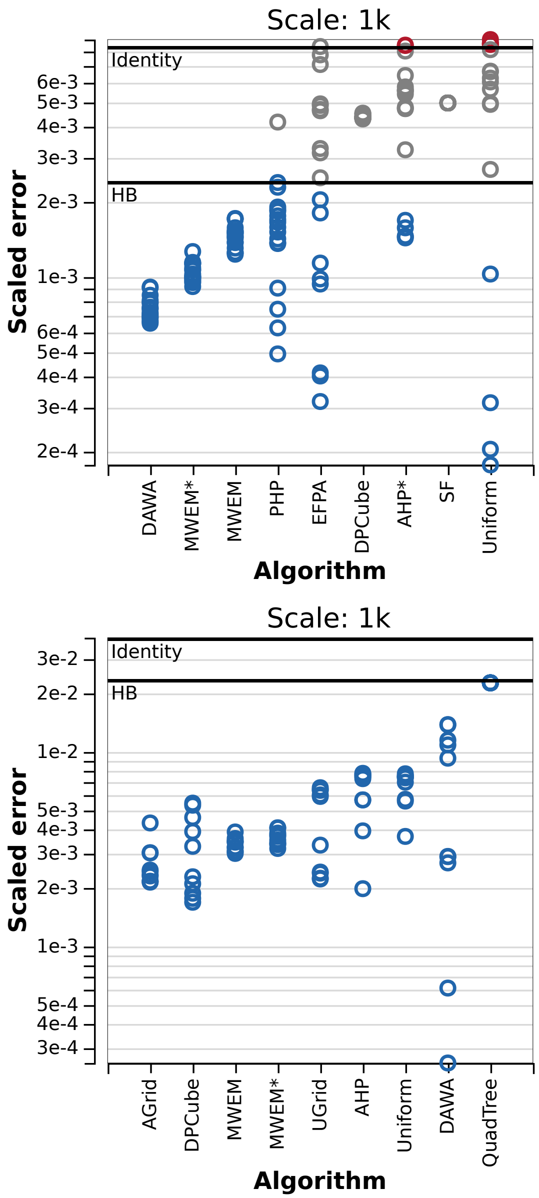


2D



1D

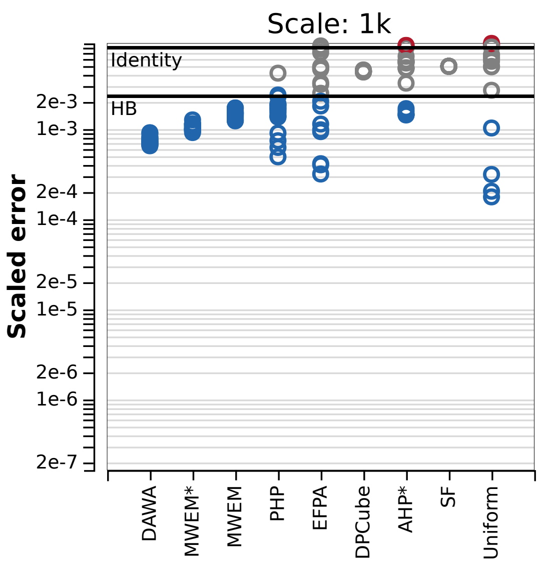
2D



Increasing scale

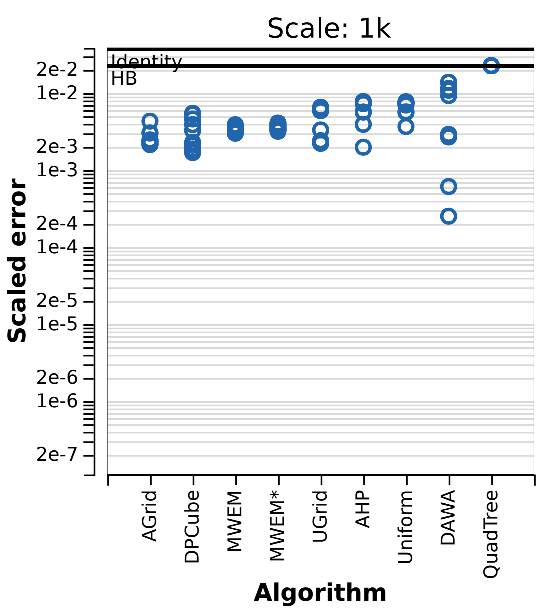
Some data-dependent algorithms fail to offer benefits at larger scales (or higher epsilons).

1D



Increasing scale

2D



Summary

- Empirical study on 1D and 2D range query workloads shows:
  - Significant variation in error for data-dependent methods
  - Significant trade-offs with “signal strength”
    - Low signal: data-dependent methods outperform
    - High signal: data-independent method outperform

Outline

1. Algorithm landscape
2. Motivating challenge: a Census workload
3. Scaling the matrix mechanism
4. Results on the Census workload
5. Data-adaptive algorithms and trade-offs
6. **Open problems**

Open problems

- Scaling to high dimensional data
  - HDMM: strategy selection is no longer bottleneck; data vector is.
    - Recent approach: measure low-dimensional projections, use graphical model techniques for global inference
    - Mis-match between strategy optimization and inference
  - Better understanding of tradeoffs between algorithmic approaches in high dimensions.

Open problems

- Beyond linear queries
  - Common SQL aggregate queries are not linear; how do we answer them effectively?

Thank you

- *Optimizing Error of High-Dimensional Statistical Queries Under Differential Privacy*. Ryan McKenna, Gerome Miklau, Michael Hay, Ashwin Machanavajjhala PVLDB 2018

- *The matrix mechanism: optimizing linear counting queries under differential privacy*. Chao Li, Gerome Miklau, Michael Hay, Andrew McGregor and Vibhor Rastogi VLDB Journal 2015

- *Principled Evaluation of Differentially Private Algorithms using DPBench.* Michael Hay, Ashwin Machanavajjhala, Gerome Miklau, Yan Chen, and Dan Zhang. SIGMOD 2016