38th International Conference on Very Large Data Bases, Istanbul, Turkey

Low-Rank Mechanism: Optimizing Batch Queries under Differential Privacy

Ganzhao Yuan[1], Zhenjie Zhang[2], Marianne Winslett[2,3], Xiaokui Xiao[4], Yin Yang[2,3], Zhifeng Hao[1]

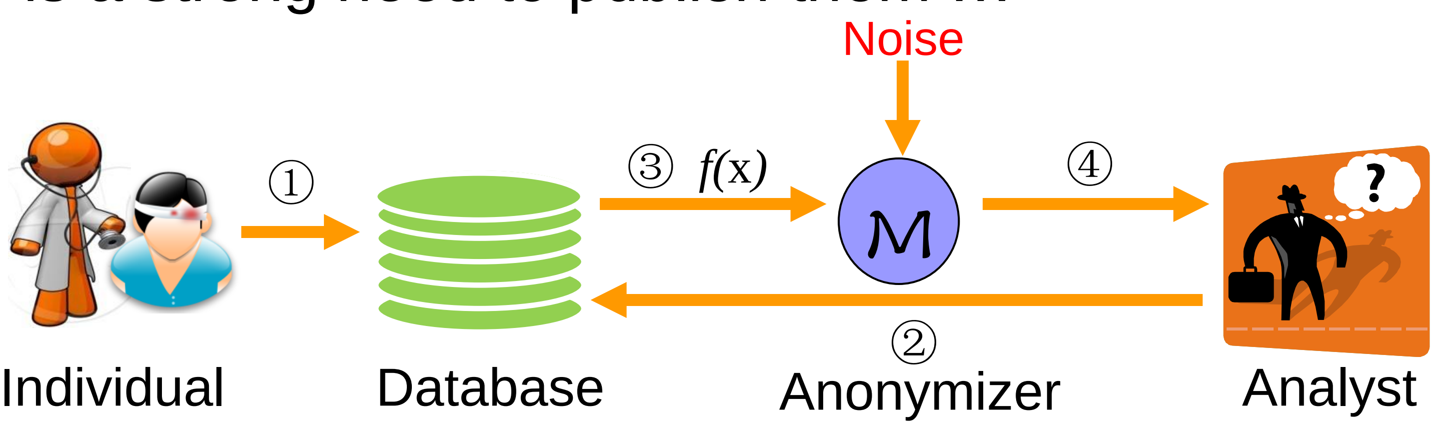
1. South China University of Technology
2. Advanced Digital Sciences Center
3. University of Illinois at Urbana-Champaign
4. Nanyang Technological University

Introduction

- **Personal information**:
  - Census data
  - Social networks
  - Medical / public health data
  - Recommendation systems records

Such data collections are of significant research value.

There is a strong need to publish them …



How to Design the Anonymizer $\mathcal{M}$?

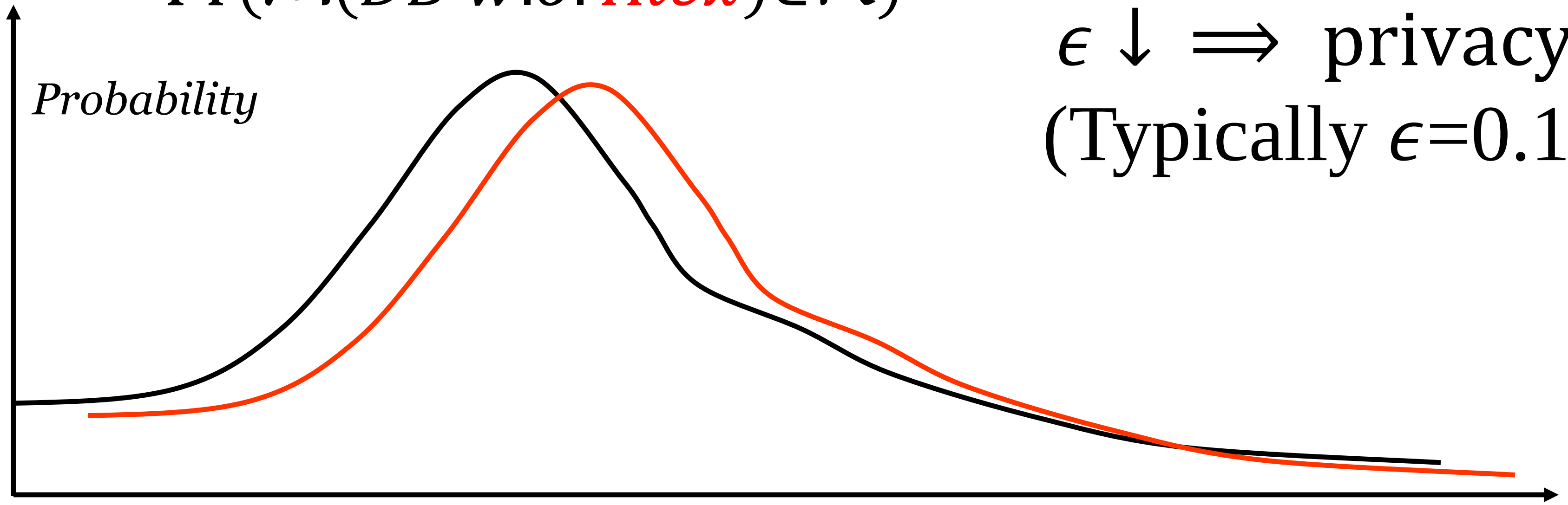
Differential Privacy [Dwork et al., TCC'06]

$\mathcal{M}$ satisfies ε–differential privacy, if for all possible DB, any individual (say, $Alex$), and all possible result set $\mathcal{R} \subseteq$ Range($\mathcal{M}$):

$$\frac{\Pr(\mathcal{M}(DB\ with\ Alex) \in \mathcal{R})}{\Pr(\mathcal{M}(DB\ w.o.\ Alex) \in \mathcal{R})} \leq e^{\epsilon} \approx 1 \pm \epsilon$$

$\epsilon \downarrow \Longrightarrow$ privacy $\uparrow$

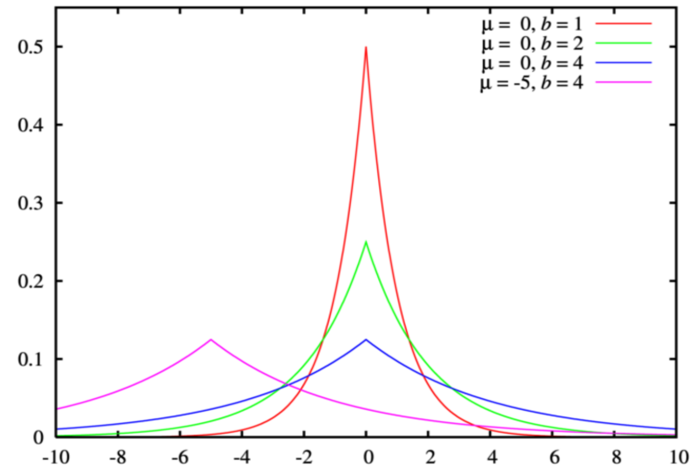
(Typically ϵ=0.1)



How to Achieve Differential Privacy

- Laplace Noise
  [Dwork et al., TCC'06]
- Gaussian Noise [Dwork et al., EUROCRYPT'12]
  - For (ε, δ)-differential privacy, a weaker differential privacy definition

Laplace Noise



pdf: $p(y) = \frac{1}{2\sigma}\exp(\frac{-||y||_1}{\sigma})$

We only focus on the Laplace Noise.

Laplace Mechanism

- Main idea: calibrate the noise according to the sensitivity of $f(DB)$, denoted as Δ_f.
- Sensitivity: For $f: D \to \mathbb{R}^m$:

$$\Delta f = \max_{DB_1, DB_2} \| f(DB_1) - f(DB_2) \|_1$$

for all DB_1, DB_2 differing in at most one record.

- **Theorem**: $\mathcal{M}(DB) = f(DB) + \mathrm{Lap}(\frac{\Delta_f}{\epsilon})^m$ satisfies ε-differential privacy.

Linear Counting Queries

Name	State	HIV+
Alice	NY	Yes
Bob	NJ	Yes
Carol	NY	Yes
Dave	CA	Yes
...		

(a) Patient records

State	# of HIV+ patients
NY	82,700
NJ	19,000
CA	67,000
WA	5,900
...	

(b) Statistics on HIV+ patients

$q_1 =$	$x_{NY} + x_{NJ} + x_{CA} + x_{WA}$
$q_2 =$	$x_{NY} + x_{NJ}$
$q_3 =$	$x_{CA} + x_{WA}$

↔

Answer **Workload Matrix:** W **Data:** D

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} x_{NY} \\ x_{NJ} \\ x_{CA} \\ x_{WA} \end{bmatrix}$$

Linear Counting Queries: Example

$q_1 =$	$x_{NY} + x_{NJ} + x_{CA} + x_{WA}$
$q_2 =$	$x_{NY} + x_{NJ}$
$q_3 =$	$x_{CA} + x_{WA}$

- Naïve solution: Noise on Result:

$$\mathcal{M}_R(\mathrm{W}, \mathrm{D}) = \mathrm{WD} + \mathrm{Lap}(\Delta_{\mathrm{W}}/\epsilon)^m$$

Sensitivity: 2

Expected error variance: $2\Delta^2/\epsilon^2 = 8/\epsilon^2$

Linear Counting Queries: Example

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} x_{NY} \\ x_{NJ} \\ x_{CA} \\ x_{WA} \end{bmatrix}$$

- Naïve solution: Noise on Data:

$$\mathcal{M}_D(\mathrm{W}, \mathrm{D}) = \mathrm{W}(\mathrm{D} + \mathrm{Lap}(1/\epsilon)^{n})$$

Sensitivity:1

Expected error variance: $8/\epsilon^2 + 4/\epsilon^2 + 4/\epsilon^2 = 16/\epsilon^2$

Linear Counting Queries: Example

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} \color{red}{x_{NY}} + x_{NJ} \\ x_{CA} + x_{WA} \end{bmatrix}$$

- Best Strategy:
  - $\mathrm{q}'_1 = x_{NY} + x_{NJ}$, $\mathrm{q}'_2 = x_{CA} + x_{WA}$
  - $q_1 = q'_1 + q'_2$, $q_2 = q'_1$, $q_3 = q'_2$

Sensitivity:1

Expected error variance: $2/\epsilon^2 + 1/\epsilon^2 + 1/\epsilon^2 = 4/\epsilon^2$

Outline of this talk

- Low-Rank Mechanism
- Optimization Algorithms for LRM
- Experimental Result
- Future Work

Low-Rank Mechanism

Low-Rank Mechanism:

- Naïve solutions:
  - Noise on Data:

    $$\mathcal{M}_D(\mathrm{W}, \mathrm{D}) = \mathrm{W}(\mathrm{D} + \mathrm{Lap}(\Delta(\mathrm{I})/\epsilon)^{n})$$

  - Noise on Result:

    $$\mathcal{M}_R(\mathrm{W}, \mathrm{D}) = \mathrm{WD} + \mathrm{Lap}(\Delta(\mathrm{W})/\epsilon)^{m}$$

- Our approach: Low-Rank Mechanism:

$$\mathcal{M}_{LRM}(\mathrm{W}, \mathrm{D}) = \mathrm{B}(\mathrm{LD} + \mathrm{Lap}(\Delta_{\mathrm{L}}/\epsilon)^{r})$$

$$\mathrm{W} \in \mathbb{R}^{m\times n}, \mathrm{B} \in \mathbb{R}^{m\times r}, \mathrm{L} \in \mathbb{R}^{r\times n}$$

$$r \leq \min(m, n)$$

Low-Rank Mechanism

$$\mathcal{M}_{LRM}(\mathrm{W}, \mathrm{D}) = \mathrm{B}(\mathrm{LD} + \mathrm{Lap}(\Delta_{\mathrm{L}}/\epsilon)^{r})$$
$$\mathrm{W} \in \mathbb{R}^{m \times n}, \mathrm{B} \in \mathbb{R}^{m \times r}, \mathrm{L} \in \mathbb{R}^{r \times n}$$

- Workload matrix decomposition: $\mathrm{W} = \mathrm{BL}$
- Noise is injected into intermediate result LD
- Expected error variance: $2\mathrm{tr}(B^T B){\Delta_{\mathrm{L}}}^2/\epsilon^2$

Minimizing Expected Error

$$\min_{B,L} \frac{2tr(B^T B){\Delta_{\mathrm{L}}}^2}{\epsilon^2} \quad \text{s.t. } W = BL$$

Where $\Delta_{\mathrm{L}} = \max_j \sum_i |L_{ij}|$

Nonsmooth!

An important observation:

The sensitivity of L is not important!

Given decomposition $W = BL$ and any positive constant α, we can always construct another decomposition $W = B'L'$ such that $\Delta_{\mathrm{L'}} = \alpha$ and

$$2tr(B^T B){\Delta_{\mathrm{L}}}^2 = 2tr(B'^T B'){\Delta_{\mathrm{L'}}}^2$$

Optimization Problem:

OP1: $\min_{B,L} \operatorname{tr}(B^T B) {\Delta_\mathrm{L}}^2$, s.t. $W = BL, \Delta_\mathrm{L} = \max_j \sum_i |L_{ij}|$

Fix the Sensitivity!

OP2: $\min_{B,L} \operatorname{tr}(B^T B)$, s.t. $W = BL, \Delta_\mathrm{L} = \max_j \sum_i |L_{ij}| = 1$

Eliminate the $\max$ operator

OP3: $\min_{B,L} \operatorname{tr}(B^T B)$, s.t. $W = BL, \forall \mathrm{j} \sum_i |L_{ij}| \leq 1$

Optimization Problem:

$$\boxed{\begin{gathered}\min_{B,L} \operatorname{tr}(B^T B) \\ \text{s.t. } W = BL, \\ \forall \mathrm{j} \sum_i |L_{ij}| \leq 1\end{gathered}}$$

How much noise do we add?

- Dwork et al. [TCC'06]: $\mathcal{O}(n^2)$
- De et al. [TCC'12]: $\mathcal{O}(\min(m, n)^2)$
- Our result : $\mathcal{O}\left(\sum_{i=1}^{k} \lambda_i^2 \, r\right)$

  - r is the rank of the workload matrix W

Optimization Algorithms

Optimization Problem:

$$\min_{B,L} \frac{1}{2} tr(B^T B)$$

$$\text{s.t. } W = BL,$$

$$\forall j \sum_i |L_{ij}| \leq 1$$

Challenge: Non-Convex, Non-Smooth

Optimization Algorithms

- For linear constraints：

Introduce a positive penalty β

& Lagrange multiplier π

$$\min_{B,L} \frac{1}{2} tr(B^T B)$$
$$\text{s.t. } W = BL,$$
$$\forall j \sum_i |L_{ij}| \leq 1$$

- For $\mathcal{L}_1$ regularized constraints：

  Projective Gradient Descent

- Augmented Lagrangian subproblem:

$$\mathcal{J}(B, L, \beta, \pi) = \frac{1}{2} tr(B^T B) + \langle \pi, W - BL \rangle$$
$$+ \frac{\beta}{2} \| \mathrm{W} - \mathrm{BL} \|_F^2, s.t. \forall \mathrm{j} \sum_i |L_{ij}| \leq 1$$

Workload Matrix Decomposition Algorithm
1. Initialize $\pi = 0^{m\times n}$, $\beta = 0$
2. Loop
3. While not converged // solve the subproblem
4. $\mathrm{B} \leftarrow \min_{B} \mathcal{J}(B, L, \beta, \pi)$ //Closed Form solution
5. $\mathrm{L} \leftarrow \min_{L} \mathcal{J}(B, L, \beta, \pi)$ //Projective Gradient Descent
6. If ($\parallel W - BL \parallel_F < \gamma$), return $\{B, L\}$
7. Increase the penalty parameter β
8. Update the Lagrange multiplier $\pi \leftarrow \pi + \beta(W - BL)$

Subproblem:

$$\mathcal{J}(B, L, \beta, \pi) = \frac{1}{2} tr(B^T B) + \langle \pi, W - BL \rangle + \frac{\beta}{2} \parallel \mathrm{W} - \mathrm{BL} \parallel_F^2, s.t. \forall \mathrm{j} \sum_i |L_{ij}| \leq 1$$

Highlights of Our Algorithms

- Inexact Augmented Lagrangian Multiplier (ALM) method for low-rank matrix completion [Lin, et al., arXiv'10]
- Updating L:
  - No Lagrangian Multiplier for the $\mathcal{L}_1$ regularized terms
  - Instead we use Nesterov's Optimal Projective Gradient Descent
- Updating B: closed form solution

Convergence Rate: Linear

- If $\{\mathrm{B}^{\mathrm{k}}, \mathrm{L}^{\mathrm{k}}\}$ is the temporary solution after the k^{th} iteration and $\{\mathrm{B}^{*}, \mathrm{L}^{*}\}$ is the optimal solution, we have

$$|\mathrm{tr}\left({\mathrm{B}^{\mathrm{k}}}^{\mathrm{T}}\mathrm{B}^{\mathrm{k}}\right) - \mathrm{tr}\left({\mathrm{B}^{*}}^{\mathrm{T}}\mathrm{B}^{*}\right)| \leq \mathcal{O}\left(\frac{1}{\beta^{\mathrm{k}-1}}\right)$$

Matrix Mechanism and Low-Rank Mechanism

- LRM is inspired by the Matrix Mechanism [Li et al., PODS'10]: $\mathcal{M}_{MM}(\mathrm{W},\mathrm{D}) = \mathrm{W}\left(\mathrm{D} + \mathrm{A}^{\dagger}\mathrm{Lap}(\Delta_{\mathrm{A}}/\epsilon)^{\,n}\right)$
  - MM looks similar to LRM: A→L, WA⁺→B
    $$\mathcal{M}_{LRM}(\mathrm{W},\mathrm{D}) = \mathrm{B}(\mathrm{LD} + \mathrm{Lap}(\Delta_{\mathrm{L}}/\epsilon)^{\,r})$$
    - Authors of MM point out that LRM can be seen as a special case of MM
  - But there is an important difference in formalization: whether or not to use pseudo inverse (i.e., A^{+})
    - The pseudo-inverse-free formalization in LRM allows more freedom in choosing optimization solutions

Matrix Mechanism and Low-Rank Mechanism

- The optimization solution in MM is inefficient
  - MM needs to solve: $\min_A \| A \|_1^2 \, tr(A^{\dagger} W^T W A^{\dagger^T})$
  - This is hard, and the solution in MM has a high computation cost
  - Alternative solution: solve: $\min_{A} \| A \|_2^2 \, tr(A^{\dagger} W^T W A^{\dagger^T})$
  - But this leads to poor result accuracy (as shown in our experiments)
- LRM avoids these problems
  - Can be seen as a refined version of MM
- New results for MM [Li and Miklau, VLDB’12]: MM successfully optimizes linear batch queries under the (ε, δ)-differential privacy definition

Experimental Results

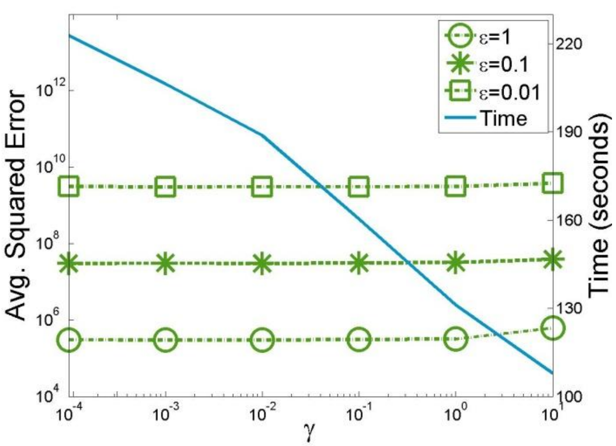
Competitors:

- Low-Rank Mechanism (LRM): [This paper]
- Matrix Mechanism (MM): [Li et al., PODS'10]
- Laplace Mechanism (LM): [Dwork et al, TCC'06]
- Wavelet Mechanism (WM): [Xiao et al, ICDE'10]
- Hierarchical Mechanism (HM) : [Hay et al, VLDB'10]

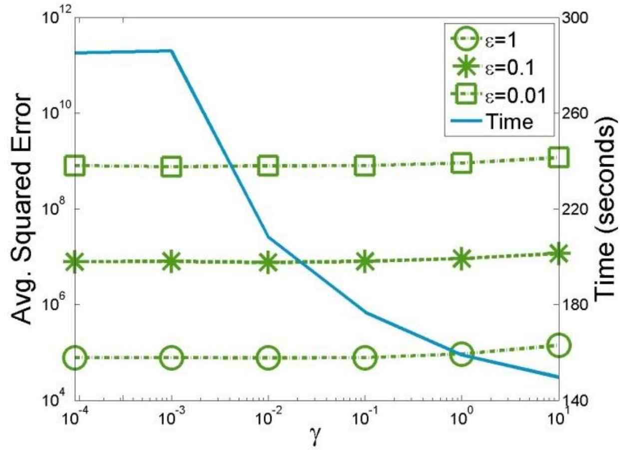
Parameters:

γ	$0.0001, 0.001, \mathbf{0.01}, 0.1, 1, 10$
r	$\{0.8, 1.0, \mathbf{1.2}, 1.4, 1.7, 2.1, 2.5, 3.0, 3.6\} \times rank(W)$
n	$128, 256, 512, \mathbf{1024}, 2048, 4096, 8192$
m	$64, 128, \mathbf{256}, 512, 1024$
s	$\{0.1, 0.2, 0.3, 0.4, \mathbf{0.5}, 0.6, 0.7, 0.8, 0.9, 1.0\} \times min(m, n)$

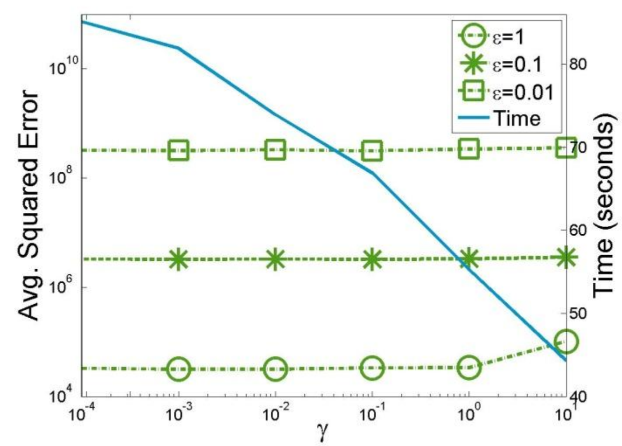
Varying γ for Low-Rank Mechnism



WDiscreate



WRange



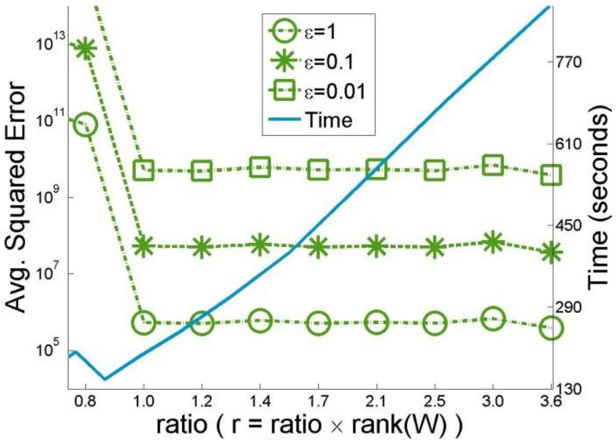
WRelated

$$\| \mathrm{W} - \mathrm{BL} \|_F^2 \leq \gamma$$

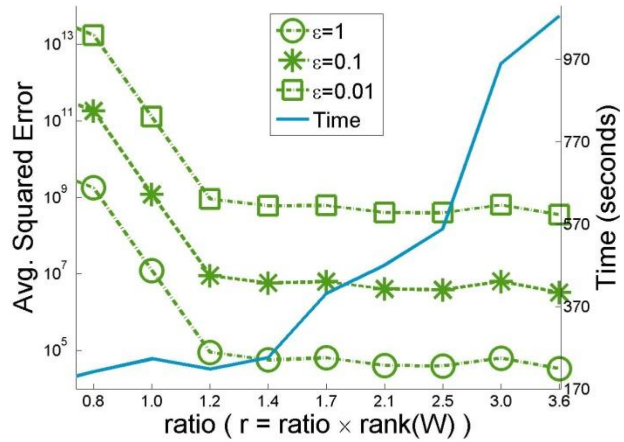
Observation:

A larger value for γ is preferred

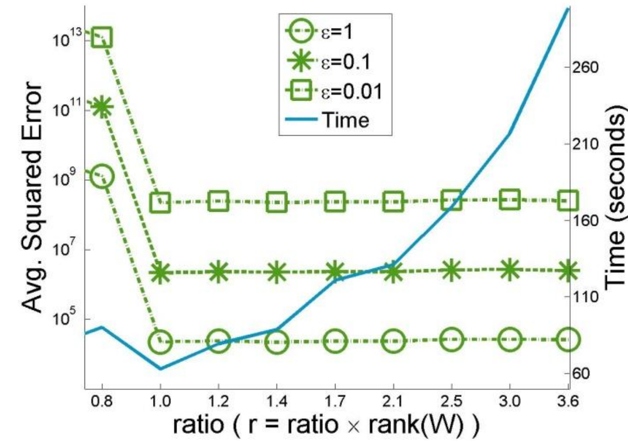
Varying r for Low-Rank Mechnism



WDiscreate



WRange



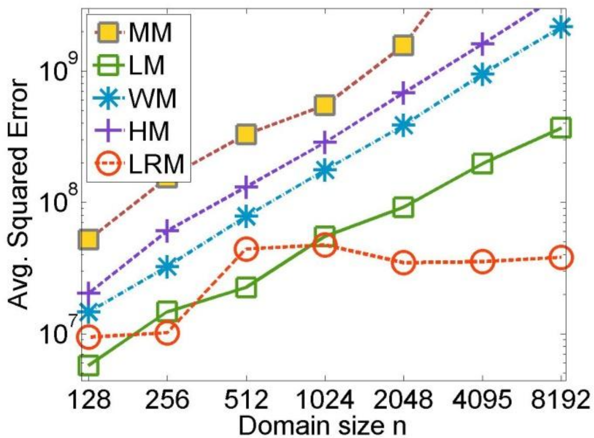
WRelated

$W \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{m\times r}, L \in \mathbb{R}^{r\times n}$

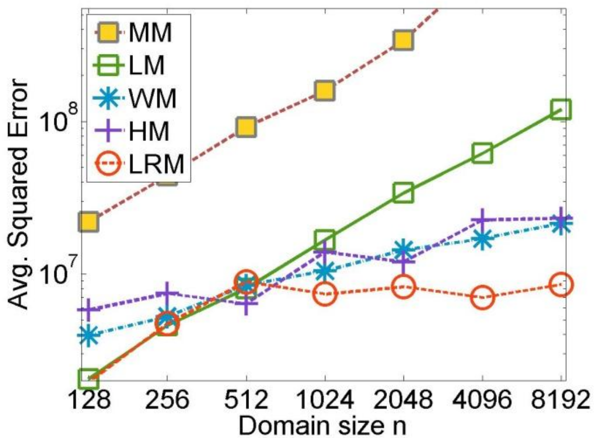
Observation:

$r = (1\sim1.2) \times rank(W)$ obtains good results.

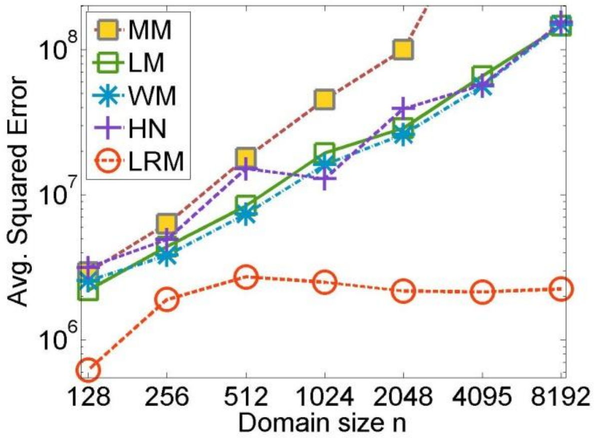
Varying n for all methods



WDiscreate



WRange

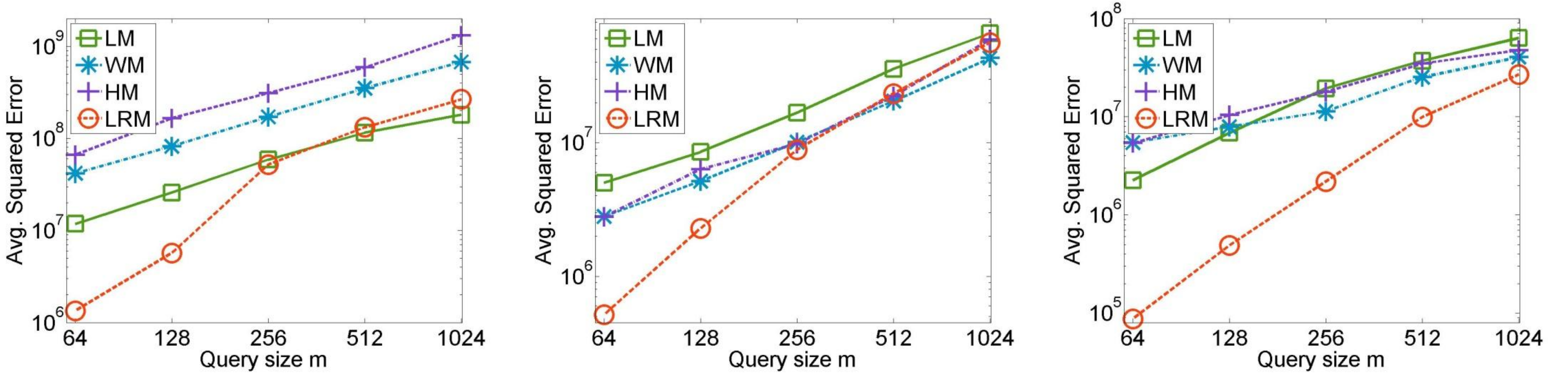


WRelated

Observations:

- Matrix Mechanism (MM) obtains poor accuracy.
- LRM's error becomes stable when the domain size exceeds 512. Moreover, It significantly outperforms other mechanisms when n is large.

Varying m for all methods

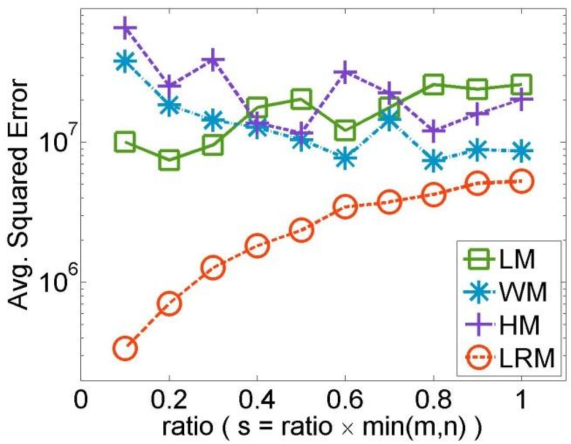


WDiscreate WRange WRelated

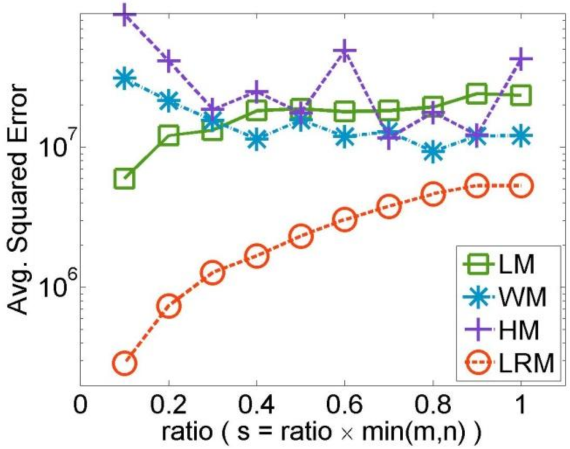
Observations:

- LRM outperforms all other mechanisms, when the number of queries $m \ll n$.
- As m grows, the performance of all mechanisms on all workloads gradually converges.

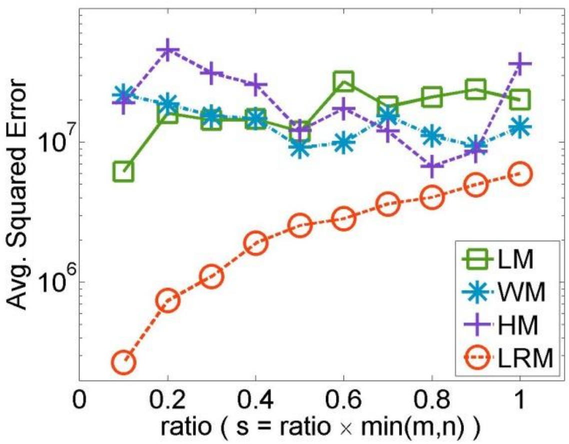
Varying s for all methods



Search Logs



NetTrace



Social Network

Observations:

$$W = AC, \qquad A \in \mathbb{R}^{m \times s}, C \in \mathbb{R}^{s \times m}$$

- LRM substantially outperforms other methods, especially when the rank of the workload matrix is low.
- With increasing rank of W, LRM's error grows.
- The low rank property is the main reason behind LRM's advantages.

Future Work

- Data-aware and workload-aware optimization
- Global sensitivity → local sensitivity
- Applications to social networks and graphs

Thank you!

Our code is available online:
http://yuanganzhao.weebly.com/

Appendix: Upper Bound and Lower Bound

- Use SVD to construct a feasible solution:

$$\mathrm{W} = \mathrm{U\Sigma V} = \sqrt{\mathrm{r}}\mathrm{U\Sigma}\left(\frac{1}{\sqrt{\mathrm{r}}}\mathrm{V}\right) = \mathrm{BL}$$

- Upper Bound$= tr(B^T B) = tr((\sqrt{\mathrm{r}}\mathrm{U\Sigma})^T (\sqrt{\mathrm{r}}\mathrm{U\Sigma}))/\epsilon^{\ 2}$

$$= \frac{tr(\Sigma^T \mathrm{U}^T \Sigma U) r}{\epsilon^{\ 2}} = \frac{\sum_{k=1}^{r} \lambda_k^2\ r}{\epsilon^{\ 2}} \leq\ \lambda_1^2 r^2 = \mathcal{O}(\mathrm{r}^2)$$

- Lower Bound $= \mathcal{O}\left(r^3 Vol(PW\mathrm{B}_1^n)^{2/r}/\varepsilon^2\right)$ [Hardt et al., STOC 2010] **(**We construct the lower error bound by SVD:$\mathrm{W} = \mathrm{U\Sigma V}$ and let $P = U^T$. $\mathrm{Vol}(\mathrm{B}_1^r) = 2^r/r!$ **)**
- Lower Bound $= \mathcal{O}\left(r^3 (2^r/r! \prod_{k=1}^{r} \lambda_k)^{2/r}/\varepsilon^2\right) \geq$

$$\mathcal{O}\left(r^3 \cdot r^{-2} \cdot (\prod_{k=1}^{r} \lambda_k)^{2/r}/\varepsilon^2\right) \geq \mathcal{O}\left(r/\varepsilon^2\right) = \mathcal{O}(r)$$