CERIAS

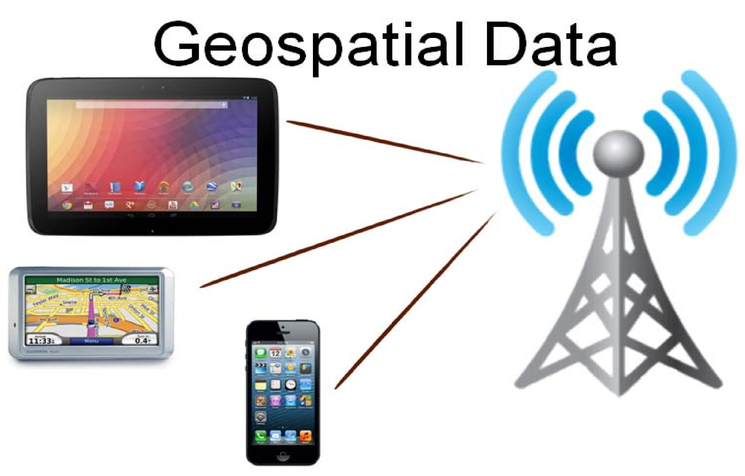
The Center for Education and Research in Information Assurance and Security

PURDUE UNIVERSITY

Differentially Private Grids for Geospatial Data

Wahbeh Qardaji, Weining Yang, Ninghui Li

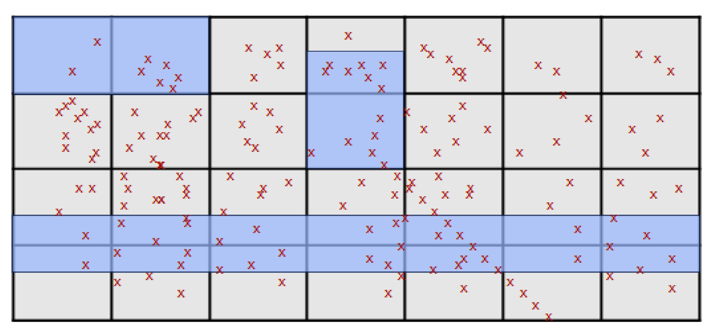
Problem Definition



- **Data**: Two dimensional. Each tuple is a point in two-dimensional space.
- **Goal**: publish a synopsis of the dataset to accurately answer range count queries.
  - Each query is a rectangle in the data domain.

Challenge: balance privacy and utility goals:

- Privacy Goal: Differential privacy
  - Achieved by adding noise to query answers
- Utility Goal: High accuracy in count queries



Differential Privacy

$$\Pr[\mathcal{A}(D)=S] \le e^{\epsilon} \cdot \Pr[\mathcal{A}(D')=S]$$
$$\forall S \in Range(\mathcal{A})$$
$$\forall D, D' : D \setminus D' = \{t\}$$

For any two datasets that differ by at most one tuple, a differentially private algorithm will behave approximately the same, i.e., no single tuple affect the output too much.

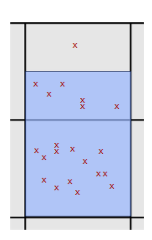
To satisfy differential privacy, we can add Laplace noise to the output.

$$\mathcal{A}_g(D) = g(D) + \mathsf{Lap}\left(\frac{\mathsf{GS}_g}{\epsilon}\right)$$
$$\text{where } \mathsf{GS}_g = \max_{(D,D'):D\simeq D'} |g(D) - g(D')|,$$
$$\text{and } \Pr[\mathsf{Lap}(\beta) = x] = \frac{1}{2\beta} e^{-|x|/\beta}$$

Error Analysis

For all the methods, one partitions the domain into cells and adds noises to counts of each cells. There are two sources of error when answering a query.

1. Noisy error: Error from Laplace noise.
2. Non-uniformity error: Error comes from the distribution of dataset.



Smaller noisy error $\Leftrightarrow$ Coarser partition

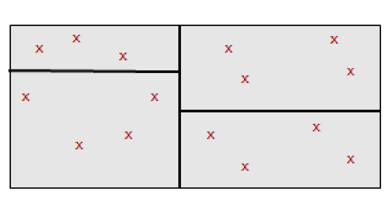
Smaller non-uniformity error $\Leftrightarrow$ Finer partition

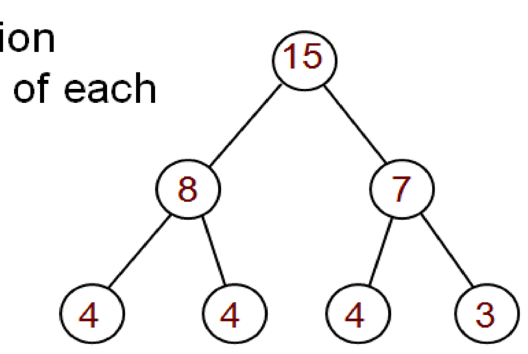
Optimization: Find a best partition

Previous Methods

KD-tree : recursively partition along the median of each axis.
Quad-tree : Recursively partition each region into 4 quadrants.
KD-hybrid : Quad-tree at first few levels and KD-tree for the other levels.
Privlet method : Apply Wavelet transformation and build a binary tree.

- KD-Trees
  - Recursively partition along the median of each axis



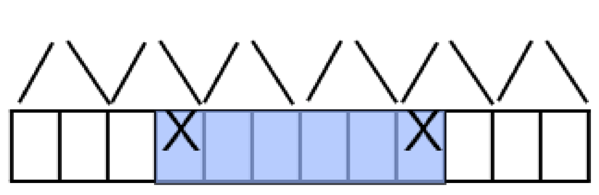


Analysis of Hierarchical Method

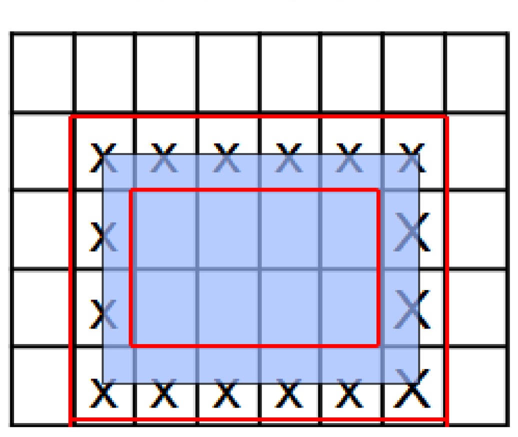
One can benefit from doing hierarchical

- Use parent nodes instead of child nodes to answer queries.

One-dimension



Two-dimension



There are M cells and branching factor is b.

- One-dimensional case: the number of children nodes used is $O(\frac{2b}{M})$
- Two-dimensional case: the number of children nodes used is $O(\frac{4\sqrt{b}}{\sqrt{M}})$

For example:
M=10000
b=4

$$\Rightarrow \quad 2\frac{b}{M} = 2\frac{4}{10000} = 0.08\% \qquad 4\frac{\sqrt{b}}{\sqrt{M}} = 4\frac{\sqrt{4}}{\sqrt{10000}} = 8\%$$

- Such proportion can be larger for higher dimension dataset.
- Hierarchical method is not suitable for high dimension dataset.

Uniform Grid

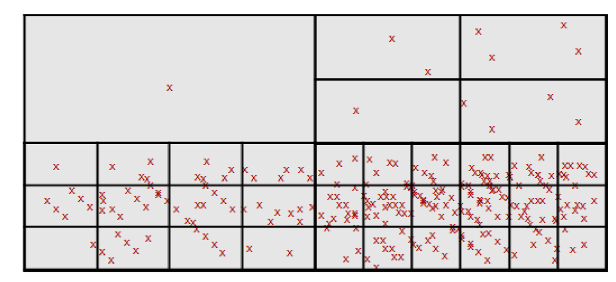
Simplest approach: partition domain into m x m cells of equal size

Standard deviation of noisy error: $\frac{\sqrt{2rm^2}}{\epsilon}$

Standard deviation of non-uniformity error: $\frac{\sqrt{r}N}{c_0 m}$

$$\arg\min_m \frac{\sqrt{2}rm}{\epsilon} + \frac{\sqrt{r}N}{mc_0} \Rightarrow m = \sqrt{\frac{N\epsilon}{c}}, c \approx 10$$

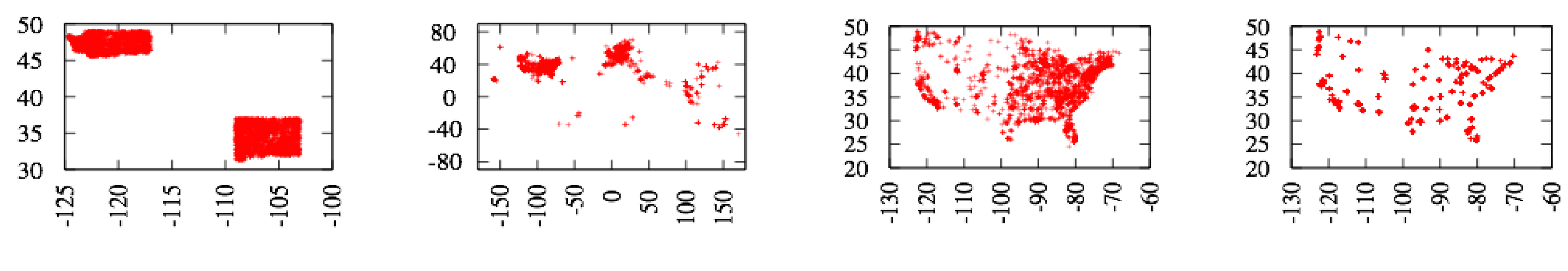
Adaptive Grid



- Idea: Adapt the level of partitioning based on the number of data points in each region.
- Two level partitioning:
  1. Lay a coarse m1 x m1 grid over the data domain and issue a DP count query
  2. Partition each into an m2 x m2 grid based on noisy count
  3. Apply constrained inference to reduce error

$$m_1 = \max\left(10, \frac{1}{4}\left\lceil\sqrt{\frac{N\epsilon}{c}}\right\rceil\right) \qquad m_2 = \left\lceil\sqrt{\frac{N'(1-\alpha)\epsilon}{c_2}}\right\rceil \qquad c_2 = \frac{c}{2} \approx 5$$

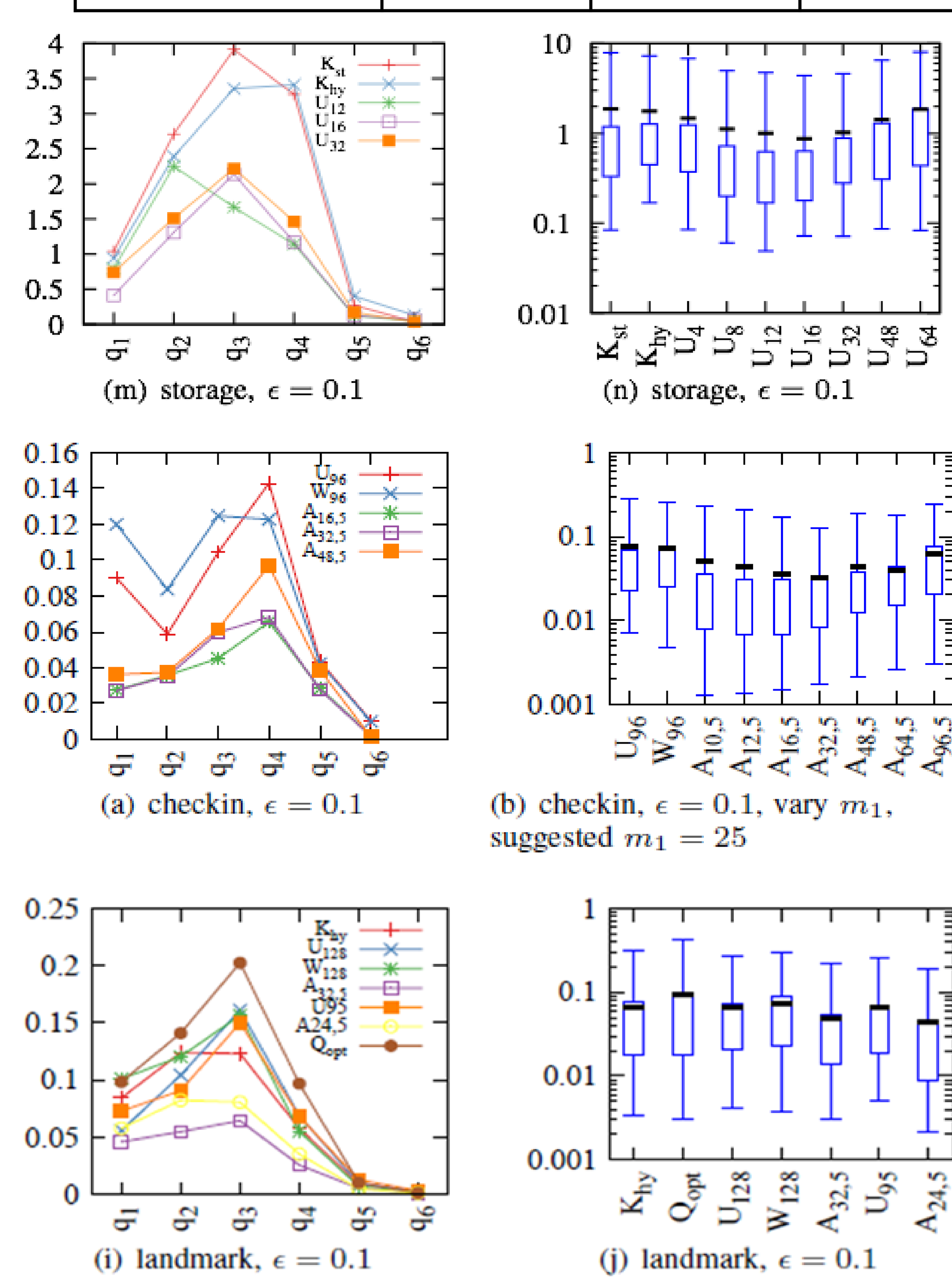
Experiment

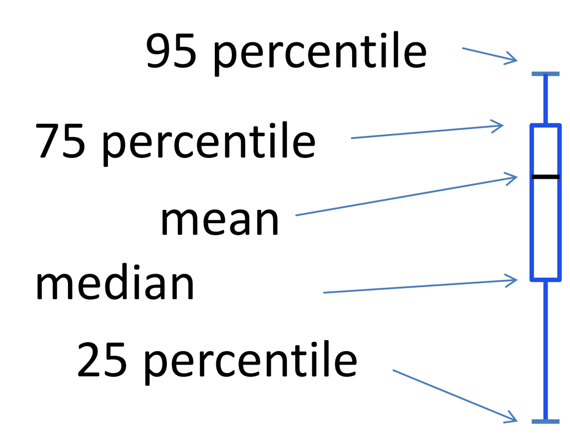


(a) The road dataset (b) The checkin dataset (c) The landmark dataset (d) The storage dataset

- We choose 6 different query sizes and for each query size, we randomly generate 200 queries.
- We measure relative error as criteria. $RE_{\mathcal{M}}(r) = \frac{|Q_{\mathcal{M}}(r) - A(r)|}{\max\{A(r), \rho\}}$

		$\epsilon = 0.1$		$\epsilon = 1.0$	
Dataset	N	Sugg. size	Best sizes	Sugg. size	Best sizes
road	1.6 M	126	48-128	400	96-192
checkin	1 M	100	64-128	316	192-384
landmark	0.9 M	95	64-128	300	256-512
storage	9 K	10	10-32	30	32-64





K_{st} : standard KD-tree.
K_{hy} : KD-hybrid
U_i: Uniform Grid with m=i
W_i: Privlet method with i by i cells.
$A_{i,j}$: Adaptive Grid with m_1=i and c_2=j
Q_{opt} : Qual-tree method with an optimized division of privacy budget

Discovery Park e-Enterprise Center