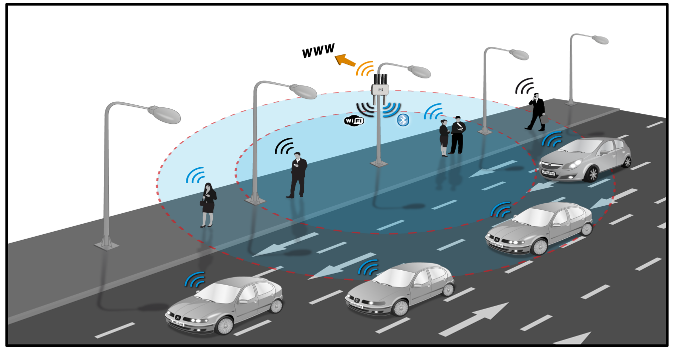
Fast & Private Spatial Range Query

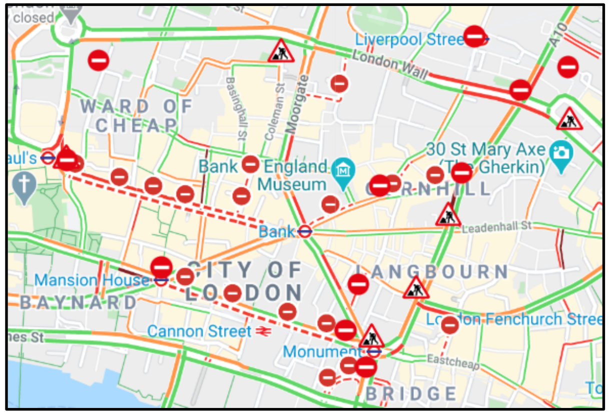
Abhirup Ghosh, Jiaxin Ding, Rik Sarkar, Jie Gao

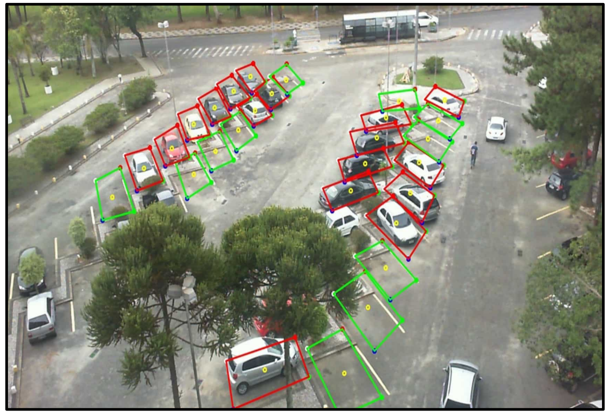
Postdoc @ Imperial

Spatial Sensing

- Densely deployed sensors collect events with locations
- Events of traffic incidents, footfall, occupancy, etc.
- Analyzing these events are important across applications

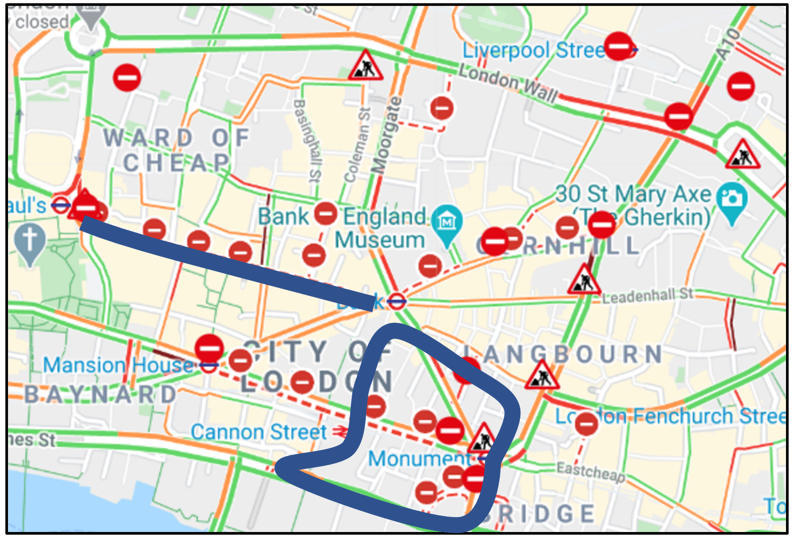


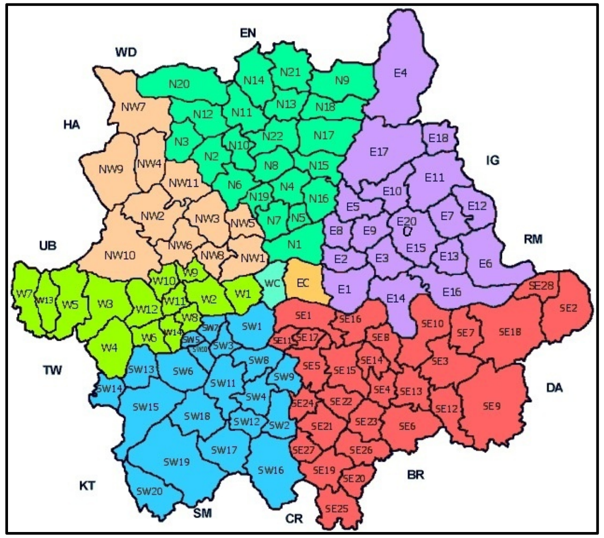




Counting range query

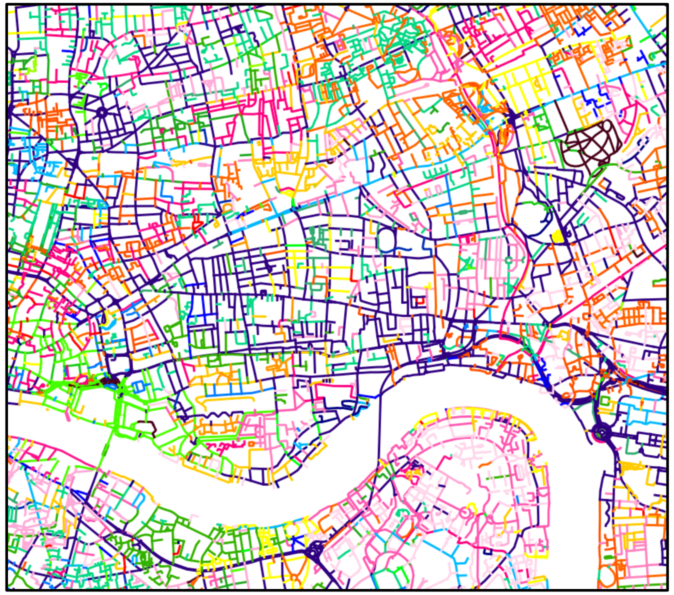
- Ranges on given spatial subdivision – map / administrative divisions
- Count events in the range
- 1D: Along a path
- 2D: Within a region





Challenges - efficiency

- Too many roads – 2.7M roads in California
- KD-tree / Quad tree do not work for path queries and non-rectangular ranges



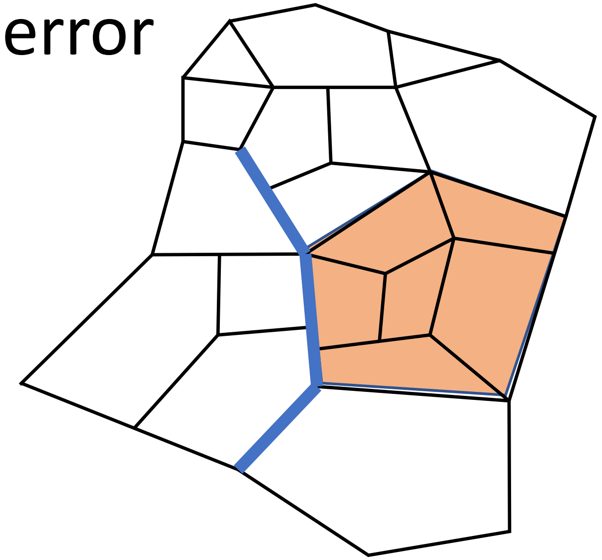
Challenges – user privacy

- Location data is sensitive
- Anonymous and aggregate counts can reveal sensitive information
- Suppose a taxi company publishes aggregate drop-offs / street
- Taxi drop-off locations correlate to home locations

Our work

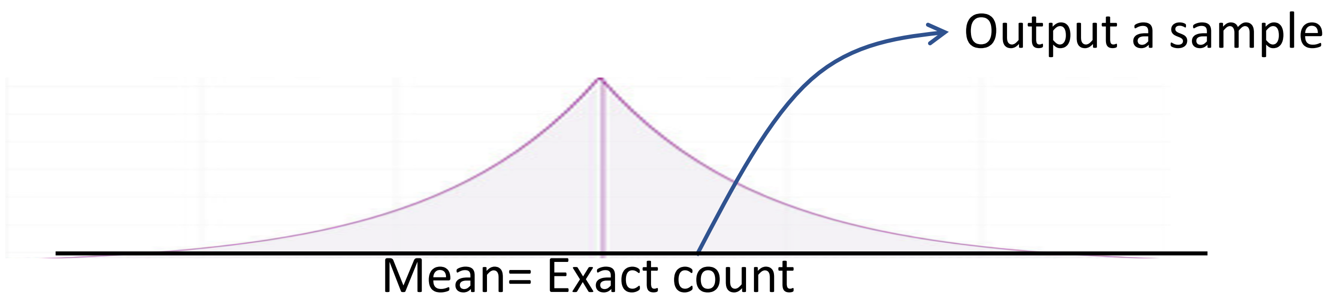
- Problem: Answer spatial range queries with privacy and efficiency
- Support un-bounded # of queries
- Planar graph to model spatial domain (road → edge; crossing → node)
- Hierarchical data structures for $O(\log n)$ query time [n = # of nodes]
- Differential Privacy with polylogarithmic error

1D queries are along shortest paths



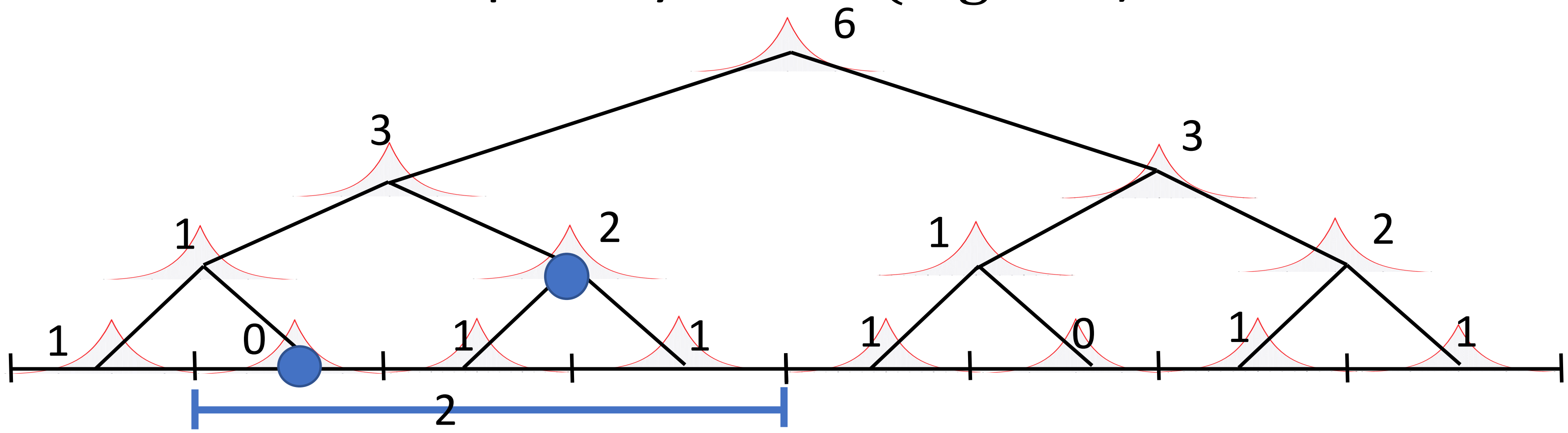
Differential Privacy for spatial range queries

- Protect occurrence of an event
- Find exact range count then add noise
- Noise $\propto$ possible # query ranges.
- # shortest paths $O(n^2)$ → large error
- Local differential privacy also adds too much noise, $O(\sqrt{n})$



Partial-sum in 1D – idea behind our method

- Build binary tree and store partial sums at tree nodes
- A range query adds noises at respective nodes
- A node can be part of O(log n) canonical ranges → noise ∝ log n
- Any query can be answered by adding O(log n) p-sums
- Achieves differential privacy with $O(\log^{1.5} n)$ error



Chan, et.al. Private and continual release of statistics, 2011

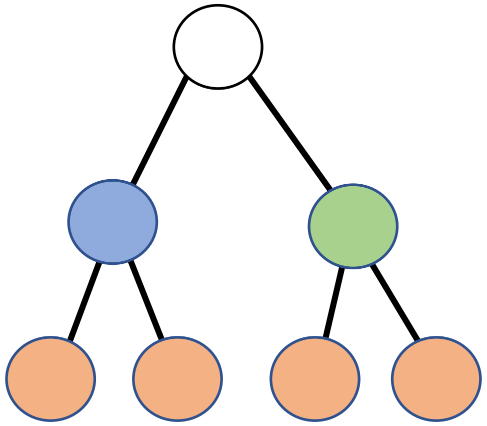
Our method: p-sum for planar graphs

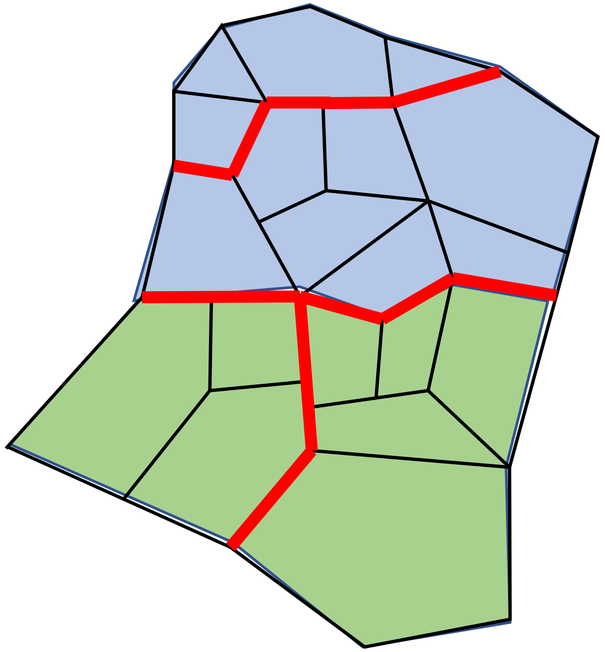
- There are $O(n^2)$ shortest paths – cannot build p-sum trees on all
- Select a few shortest paths as canonical paths
- Build p-sum trees on canonical paths

- Build two hierarchies
  - Planar separator hierarchy
  - Random sampling hierarchy

Planar separator hierarchy

- A shortest path can divide a planar graph into two balanced parts
- Efficient ways exist to find a separator (Classical result from Tarjan)
- Divide recursively – build separator hierarchy
- Separator paths are canonical paths

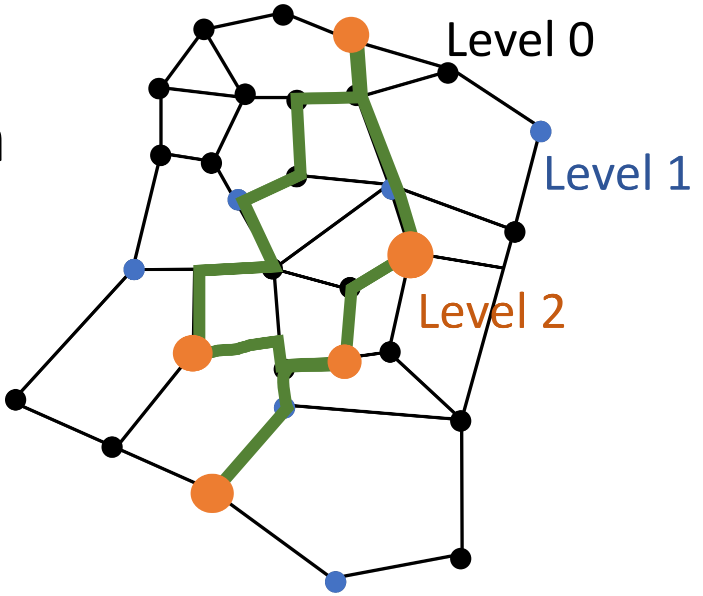




Random sampling hierarchy at separator nodes

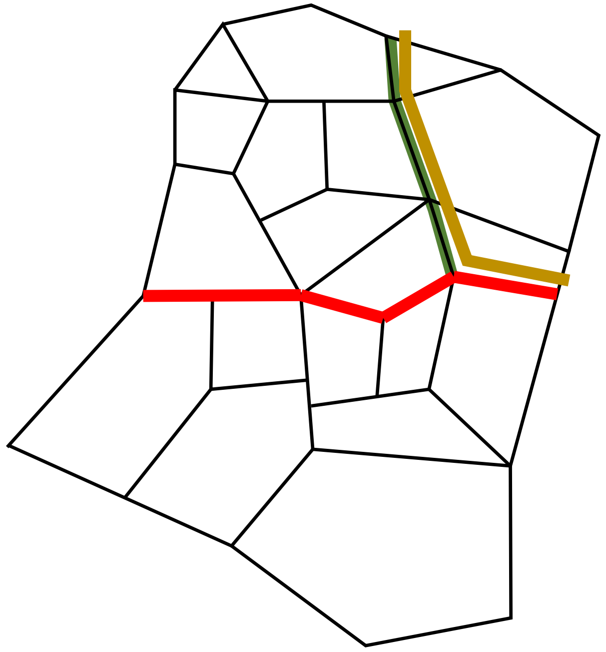
- Similar to skip list in graphs.
- All nodes are at level 0
- Independently promote a node to next level with probability ½
- Continue iteratively

- Canonical paths: shortest paths between nodes at same level



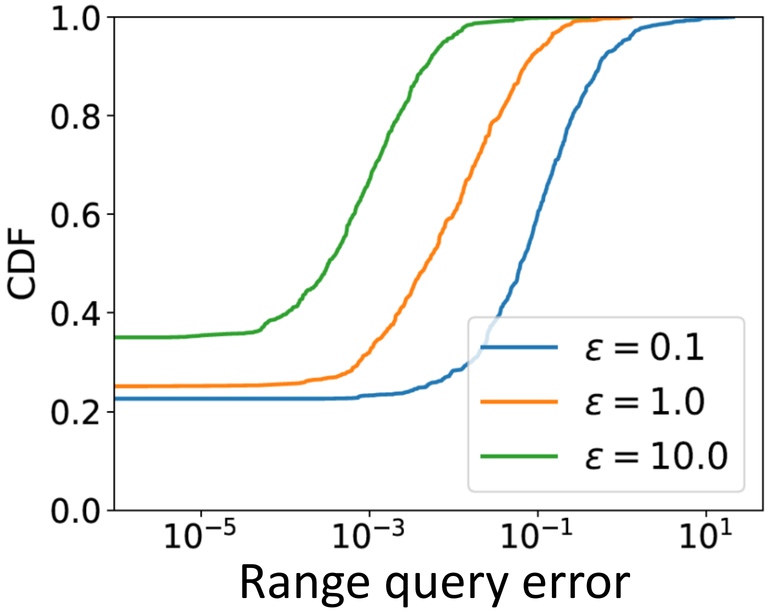
The final data structure

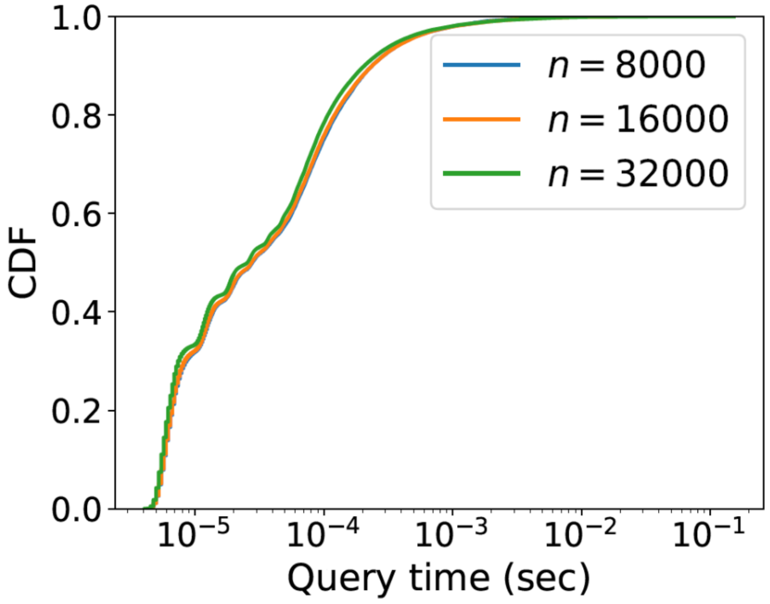
- Two types of canonical paths –
  - Type I: Separator paths
  - Type II: paths between same random level
- Build p-sum trees on all canonical paths
- Pre-compute noise samples at p-sum nodes
- Given a query path, find canonical (noisy) p-sums



Evaluate on taxi pickups event in Porto

- The road network graph is from openstreet map
- Range query error increases with decreasing ε (better privacy)
- Queries are fast – 90% queries take < 1 msec





Summary

- Fast and Private range query on Spatial data
- Select $O(n \log^2 n)$ shortest paths as canonical paths s.t. –
  - Any shortest path is a concatenation of O(log n) canonical paths
  - An edge is part of at most $O(\log^4 n)$ canonical paths
- Differential privacy with $O(\log^{4.5} n)$ error vs $O(\sqrt{n})$ in local DP

- Extends to 2D non-rectangular query ranges using differential form
- Adapts to distributed processing