Differentially Private Online Task Assignment in Spatial Crowdsourcing: A Tree-based Approach

Qian Tao¹, Yongxin Tong¹, Zimu Zhou², Yexuan Shi¹, Lei Chen³, Ke Xu¹
Outline

- Background and Motivation
- Problem Definition
- A Tree-based Framework
- Random Walk Acceleration
- Experimental Evaluation
- Conclusions
Background

- Spatial Crowdsourcing has penetrated in our life
Background

- Spatial Crowdsourcing has penetrated in our life
- Privacy leakage draws attraction in recent years

Taxi Calling  Food Delivery  Logistics

Uber  DoorDash  UPS

Uber announces new data breach affecting 57 million riders and drivers

DoorDash confirms data breach affected 4.9 million customers, workers and merchants

Zack Whittaker @zackwhittaker / 4:21 am C

UPS Reveals Data Breach
POS Malware Compromises 105,000 Transactions at 51 Stores
Mathew J. Schwartz (twitter: eurolinfsec) • August 21, 2014
A core operation of spatial crowdsourcing is task assignment.

<table>
<thead>
<tr>
<th>Type</th>
<th>Applications</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taxi Calling</td>
<td>Uber</td>
<td>Assign taxi-calling orders to drivers</td>
</tr>
<tr>
<td>Food Delivery</td>
<td>GRUBHUB</td>
<td>Assign food orders to proper deliverers</td>
</tr>
<tr>
<td>Logistics</td>
<td>UPS</td>
<td>Assign delivery tasks to proper workers</td>
</tr>
</tbody>
</table>
Background
Background

- How to make **effective task assignment** while protecting the **location privacy** of the tasks and workers?
Limitations of Existing Works

- Ignore a widely-researched and practical objective: minimizing total distance
- Lack of theoretical analysis of the effectiveness of the task assignment

H. To et al, Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server. In ICDE 2018.

Outline

- Background and Motivation
- Problem Definition
- A Tree-based Framework
- Random Walk Acceleration
- Experimental Evaluation
- Conclusions
Problem Definition

- Crowd worker $w$
  - $(x_w, y_w)$: location of the worker $w$
- Spatial task (Dynamically appears)
  - $(x_t, y_t)$: location of the task $t$
Given a set of workers \( W \) and a set of dynamically appearing tasks \( T \), we aim to design a privacy mechanism \( M \) such that:

- The mechanism guarantees the indistinguishability of the locations.
- The mechanism enables matching algorithms with minimum total distance.

The Privacy-preserving Online Minimum Bipartite Matching Problem is as follows.
Problem Definition: Indistinguishability

We require a mechanism that satisfies Geo-Indistinguishability.

A mechanism is Geo-Indistinguishable on metric space \( \mathcal{X} \) if for any \( x_1, x_2 \in \mathcal{X} \) and \( z \in \mathcal{Z} \), where \( \mathcal{Z} \) is the projection space,

\[
\mathcal{M}(x_1)(z) \leq e^{\epsilon d_X(x_1, x_2)} \mathcal{M}(x_2)(z).
\]

\( \mathcal{M}(x)(z) \): Probability of \( x \) projected to point \( z \)

\( x_1 \) and \( x_2 \) cannot be distinguished with high probability
Problem Definition

The Privacy-preserving Online Minimum Bipartite Matching Problem is as follows.

POMBM Problem

Given a set of workers $W$ and a set of dynamically appearing tasks $T$, we aim to design a privacy mechanism $M$ such that

- The mechanism guarantees the Indistinguishability of the locations
- The mechanism enables matching algorithms with minimum total distance

$$\min \sum_{(w,t) \in M} \text{dis}(w, t)$$

Make effective task assignment

Location protection from server
Outline

- Background and Motivation
- Problem Definition
- A Tree-based Framework
- Random Walk Acceleration
- Experimental Evaluation
- Conclusions
Our solution is devised based on a tree-based framework.

1. Construct and publish the HST
2. Add noise by our mechanism
3. Publish the obfuscated locations
4. Assign tasks/workers
The Well-Separated Tree is a tree space $\mathcal{T} = (V_T, d_T)$ embedded from an arbitrary space $(V, d)$ such that:

- Each leaf node corresponds to a point in $V$.
- The distance on the tree from a node at level $i$ to its parent is $2^{i+1}$. 

Hierarchical Well-Separated Tree (HST)
HST Construction

- Augment the HST to a complete one by adding fake nodes.

The original HST

The complete HST
Our solution is devised based on a tree-based framework.

- Construct and publish the HST
- Add noise by our mechanism
- Publish the obfuscated locations
- Assign tasks/workers
Main Idea: Project the exact location to one of the leaf nodes such that Geo-I is satisfied.

A mechanism is Geo-Indistinguishability on metric space $\mathcal{X}$ if for any $x_1, x_2 \in \mathcal{X}$ and $z \in \mathcal{Z}$,

$$\mathcal{M}(x_1)(z) \leq e^{\varepsilon d_{\mathcal{X}}(x_1, x_2)} \mathcal{M}(x_2)(z).$$

How to determine the probability that the exact location is projected to the leaf nodes?
Privacy Mechanism Design

- Assign different projection weights to leaf nodes based on the distance to the exact node.

Observation:
- Leaf nodes’ distance to $o_1$ depends on their least common ancestor with $o_1$.

<table>
<thead>
<tr>
<th>nodes</th>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>0</td>
</tr>
<tr>
<td>$f_1$</td>
<td>4</td>
</tr>
<tr>
<td>$f_2 - f_3$</td>
<td>12</td>
</tr>
<tr>
<td>$o_2, f_4 - f_6$</td>
<td>28</td>
</tr>
<tr>
<td>$o_3 - o_4, f_7 - f_{12}$</td>
<td>60</td>
</tr>
</tbody>
</table>
Privacy Mechanism Design

- Key Point: Assign different projection weights to leaf nodes based on the distance to the exact node.

\[ wt_i = e^{-d_i \epsilon} = e^{(4-2^{i+2})\epsilon} \]

<table>
<thead>
<tr>
<th>nodes</th>
<th>distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_1: o_1 )</td>
<td>0</td>
</tr>
<tr>
<td>( L_2: f_1 )</td>
<td>4</td>
</tr>
<tr>
<td>( L_3: f_2 - f_3 )</td>
<td>12</td>
</tr>
<tr>
<td>( L_4: o_2, f_4 - f_6 )</td>
<td>28</td>
</tr>
<tr>
<td>( L_5: o_3-o_4, f_7-f_{12} )</td>
<td>60</td>
</tr>
</tbody>
</table>

\( L_i \): the set of leaf nodes whose LCA with the exact node is located at level \( i \)

- \( L_i \) contains \( c^{i-1}(c-1) \) nodes exactly
- The distance between the exact node and nodes in \( L_i \) is \( d_i = 2^{i+2} - 4 \)
Privacy Mechanism Design

- Key Point: Assign different projection weights to leaf nodes based on the distance to the exact node.

\[ wt_i = e^{-d_i \epsilon} = e^{(4-2^{i+2})\epsilon} \]

\[ Pr_i = \frac{wt_i}{WT} \]

The probability of a node in \( L_i \) being projected to

\[ WT = 1 + \sum_{i=1}^{D} c^{i-1}(c - 1)wt_i \]

Total weight of all leaf nodes

\( L_i \): the set of leaf nodes whose LCA with the exact node is located at level \( i \)
Privacy Mechanism Design

- An Example

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Distance</th>
<th>Weights</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>(L_0: o_1)</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(L_1: f_1)</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(L_2: f_2 - f_3)</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(L_3: o_2, f_4 - f_6)</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(L_4: o_3 - o_4, f_7 - f_{12})</td>
<td>60</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ wt_i = e^{(4-2i+2)\epsilon} \]

\[ Pr_i = \frac{wt_i}{WT} \]
Privacy Mechanism Design

An Example

\( \epsilon = 1 \)

<table>
<thead>
<tr>
<th>nodes</th>
<th>weights</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_0: o_1 )</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>( L_1: f_1 )</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>( L_2: f_2 - f_3 )</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>( L_3: o_2, f_4 - f_6 )</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>( L_4: o_3 - o_4, f_7 - f_{12} )</td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>

\[ \begin{align*}
\text{weights} &= e^{(4-2i^2)\epsilon} \\
Pr_i &= \frac{wt_i}{WT}
\end{align*} \]
Privacy Mechanism Design

- An Example

\[ \epsilon = 1 \]

<table>
<thead>
<tr>
<th>nodes</th>
<th>distance</th>
<th>weights</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_0: o_1 )</td>
<td>( e^{-4} )</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( L_1: f_1 )</td>
<td>4</td>
<td>0.670</td>
<td></td>
</tr>
<tr>
<td>( L_2: f_2 - f_3 )</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( L_3: o_2, f_4 - f_6 )</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( L_4: o_3, o_4, f_7 - f_{12} )</td>
<td>60</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ wt_i = e^{(4-2i+2)\epsilon} \]

\[ Pr_i = \frac{wt_i}{WT} \]
Privacy Mechanism Design

- An Example

\[ \epsilon = 1 \]

<table>
<thead>
<tr>
<th>nodes</th>
<th>distance</th>
<th>weights</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_0: o_1 )</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( L_1: f_1 )</td>
<td></td>
<td>0.670</td>
<td></td>
</tr>
<tr>
<td>( L_2: f_2 - f_3 )</td>
<td>12</td>
<td>0.301</td>
<td></td>
</tr>
<tr>
<td>( L_3: o_2, f_4 - f_6 )</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( L_4: o_3 - o_4, f_7 - f_{12} )</td>
<td>60</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ wt_i = e^{(4 - 2i + 2)\epsilon} \]

\[ Pr_i = \frac{wt_i}{WT} \]
Privacy Mechanism Design

● An Example

$\epsilon = 1$

<table>
<thead>
<tr>
<th>nodes</th>
<th>distance</th>
<th>weights</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_0: o_1$</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$L_1: f_1$</td>
<td>4</td>
<td>0.670</td>
<td></td>
</tr>
<tr>
<td>$L_2: f_2 - f_3$</td>
<td></td>
<td>$e^{-28}$</td>
<td>0.301</td>
</tr>
<tr>
<td>$L_3: o_2, f_4 - f_6$</td>
<td>28</td>
<td>0.061</td>
<td></td>
</tr>
<tr>
<td>$L_4: o_3, o_4, f_7 - f_{12}$</td>
<td>60</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$wt_i = e^{(4-2i+2)\epsilon}$

$Pr_i = \frac{wt_i}{WT}$
Privacy Mechanism Design

● An Example

$$\epsilon = 1$$

<table>
<thead>
<tr>
<th>nodes</th>
<th>distance</th>
<th>weights</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$L_0: o_1$$</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$$L_1: f_1$$</td>
<td>4</td>
<td>0.670</td>
<td></td>
</tr>
<tr>
<td>$$L_2: f_2 - f_3$$</td>
<td>12</td>
<td>0.301</td>
<td></td>
</tr>
<tr>
<td>$$L_3: o_2, f_4 - f_6$$</td>
<td>$e^{-60}$</td>
<td>0.061</td>
<td></td>
</tr>
<tr>
<td>$$L_4: o_3-o_4, f_7-f_{12}$$</td>
<td>60</td>
<td>0.002</td>
<td></td>
</tr>
</tbody>
</table>

$$wt_i = e^{(4-2i+2)\epsilon}$$

$$Pr_i = \frac{wt_i}{WT}$$
Privacy Mechanism Design

An Example

\( \epsilon = 1 \)

<table>
<thead>
<tr>
<th>nodes</th>
<th>distance</th>
<th>weights</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L_0: o_1 )</td>
<td>0</td>
<td>1</td>
<td>0.394</td>
</tr>
<tr>
<td>( L_1: f_1 )</td>
<td>4</td>
<td>0.670</td>
<td>0.264</td>
</tr>
<tr>
<td>( L_2: f_2 - f_3 )</td>
<td>12</td>
<td>0.301</td>
<td>0.119</td>
</tr>
<tr>
<td>( L_3: o_2, f_4 - f_6 )</td>
<td>28</td>
<td>0.061</td>
<td>0.024</td>
</tr>
<tr>
<td>( L_4: o_3 - o_4, f_7 - f_{12} )</td>
<td>60</td>
<td>0.002</td>
<td>0.001</td>
</tr>
</tbody>
</table>

\[ WT = 1 + \sum_{i=1}^{4} 2^{i-1} \cdot wt_i = 2.532 \]

\[ wt_i = e^{(4-2^{i+2})\epsilon} \]

\[ Pr_i = \frac{wt_i}{WT} \]
Privacy Mechanism Design

● An Example

$$\epsilon = 1$$

<table>
<thead>
<tr>
<th>nodes</th>
<th>distance</th>
<th>weights</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_0: o_1$</td>
<td>0</td>
<td>1</td>
<td>0.394</td>
</tr>
<tr>
<td>$L_1: f_1$</td>
<td>4</td>
<td>0.670</td>
<td>0.264</td>
</tr>
<tr>
<td>$L_2: f_2 - f_3$</td>
<td>12</td>
<td>0.301</td>
<td>0.119</td>
</tr>
<tr>
<td>$L_3: o_2, f_4 - f_6$</td>
<td>28</td>
<td>0.061</td>
<td>0.024</td>
</tr>
<tr>
<td>$L_4: o_3-o_4, f_7-f_{12}$</td>
<td>60</td>
<td>0.002</td>
<td>0.001</td>
</tr>
</tbody>
</table>

$$wt_i = e^{(4-2i^2)\epsilon}$$

$$Pr_i = \frac{wt_i}{WT}$$
Privacy Mechanism Design

- **Proof of Geo-Indistinguishability**

Geo-Indistinguishability:
A mechanism is Geo-Indistinguishability on metric space $\mathcal{X}$ if for any $x_1, x_2 \in \mathcal{X}$ and $z \in \mathcal{Z}$,
\[
\mathcal{M}(x_1)(z) \leq e^{\epsilon d_{\mathcal{X}}(x_1, x_2)} \mathcal{M}(x_2)(z).
\]

Given $\mathcal{M}(x_1)(z) = Pr_{lvl}(x_1, z)$ and $\mathcal{M}(x_2)(z) = Pr_{lvl}(x_2, z)$,

\[
wt_i = e^{(4-2i+2)\epsilon}
\]

\[
Pr_i = \frac{wt_i}{WT}
\]

\[
lvl(a, b): \text{level of the least common ancestor of } a \text{ and } b
\]

\[
lvl(x_1, z) = 4
\]

\[
\mathcal{M}(x_1)(z) = Pr_{lvl}(x_1, z)
\]

\[
\mathcal{M}(x_2)(z) = Pr_{lvl}(x_2, z)
\]
Case 1: \( \text{lvl}(x_1, z) > \text{lvl}(x_1, x_2) \)

\[ \Rightarrow \text{lvl}(x_2, z) = \text{lvl}(x_1, z) \]

\[ \Rightarrow \mathcal{M}(x_1)(z) = \mathcal{M}(x_2)(z) \]

\[ \Rightarrow \text{Case 1 proved} \]
Privacy Mechanism Design

- Proof of Geo-Indistinguishability

**Geo-Indistinguishability**

A mechanism is Geo-Indistinguishability on metric space $X$ if for any $x_1, x_2 \in X$ and $z \in \mathcal{Z}$,

$$\mathcal{M}(x_1)(z) \leq e^{\epsilon d_X(x_1, x_2)} \mathcal{M}(x_2)(z).$$

**Case 2:** $\text{lvl}(x_1, z) \leq \text{lvl}(x_1, x_2)$

$$\Rightarrow \text{lvl}(x_2, z) \leq \text{lvl}(x_1, x_2)$$

$$\mathcal{M}(x_1)(z) / \mathcal{M}(x_2)(z)$$

$$= \Pr_{\text{lvl}(x_1,z)} / \Pr_{\text{lvl}(x_2,z)}$$

$$= e^{(2^{\text{lvl}(x_2,z)+2} - 2^{\text{lvl}(x_1,z)+2}) \epsilon}$$

$$\leq e^{(2^{\text{lvl}(x_1,x_2)+2} - 2^2 \epsilon}$$

$$= e^{dT(x_1,x_2) \epsilon}$$

$$\Rightarrow \text{Case 2 proved}$$

\[
wt_i = e^{(4-2^{i+2}) \epsilon} \quad \text{and} \quad Pr_i = \frac{wt_i}{WT}
\]
Tree-based Framework

- Our solution is devised based on a tree-based framework.

① Construct and publish the HST
② Add noise by our mechanism
③ Publish the obfuscated locations
④ Assign tasks/workers

Server

Task

Worker

Perturbed Task

Perturbed Worker
Main Idea: Devise a greedy algorithm on the HST.

1. Construct and publish the HST
2. Add noise by our mechanism
3. Publish the obfuscated locations
4. Assign tasks/workers
Task Assignment

- Analysis of HST-based Greedy:
  - The competitive ratio of the Tree-based Framework can be bounded by

\[
\frac{M_{TBF}}{M_{OPT}} = O\left(\frac{1}{\epsilon^4} \log N \log^2 k\right)
\]

<table>
<thead>
<tr>
<th>$N$: Number of truth nodes in HST</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$: Number of tasks/workers</td>
</tr>
<tr>
<td>$\epsilon$: Privacy budget</td>
</tr>
</tbody>
</table>

The Matching with server unknowing truth locations

The Optimal Matching even knowing all truth locations

An extra product related to privacy budget $\epsilon$

The competitive ratio of HST-Greedy without privacy
Outline

- Background and Motivation
- Problem Definition
- A Tree-based Framework
- Random Walk Acceleration
- Experimental Evaluation
- Conclusions
Privacy Mechanism Revisited

- It takes $O(c^D)$ to enumerate the probability of all leaf nodes.

**c**: Number of branches of the HST

**D**: Levels of the HST

---

**Question:** How to accelerate the mechanism?
A Random Walk Acceleration

- **Main Idea:**
  - Start from the **exact node** and randomly walk up or down with some probability at each node
  - Repeat until another **leaf node** is reached

\[
pu_0 = 0.606
\]

\[
pu_1 = 0.564
\]

\[
pu_2 = 0.304
\]

\[
pu_3 = 0.075
\]

\[
pu_4 = 0
\]
A Random Walk Acceleration

- **Algorithm Details:**
  - Phase I: Walk up until obtain a tail from the coin (at level $k$) with its head probability

$$pu_k = \frac{tw_{k+1}}{tw_k}$$

The total weights of leaf nodes outside level $k$ (including level $k$)

$$tw_k = \left\{ \begin{array}{ll}
\sum_{i=k}^{D} c^{i-1}(c-1)wt_i, & \text{if } k > 0 \\
 w_0 + \sum_{i=1}^{D} c^{i-1}(c-1)wt_i, & \text{if } k = 0
\end{array} \right.$$
A Random Walk Acceleration

- **Algorithm Details:**
  - Phase I: Walk up until obtain a tail from the coin (at level $k$) with its head probability

  $$pu_k = \frac{tw_{k+1}}{tw_k}$$

  $$tw_k = \begin{cases} 
  \sum_{i=k}^{D} c^{i-1}(c-1)wt_i, & \text{if } k > 0 \\
  w_0 + \sum_{i=1}^{D} c^{i-1}(c-1)wt_i, & \text{if } k = 0 
  \end{cases}$$

  - The total weights of leaf nodes outside level $k$ (including level $k$)
A Random Walk Acceleration

- Algorithm Details:
  - Phase I: Walk up until obtain a tail from the coin (at level $k$) with its head probability

$$p_{u_k} = \frac{tw_{k+1}}{tw_k}$$

The total weights of leaf nodes outside level $k$ (including level $k$)

- $pu_4 = 0$
- $pu_3 = 0.075$
- $pu_2 = 0.304$
- $pu_1 = 0.564$
- $pu_0 = 0.606$

$$tw_k = \begin{cases} 
\sum_{i=k}^{D} c^{i-1}(c-1)wt_i, & \text{if } k > 0 \\
 w_0 + \sum_{i=1}^{D} c^{i-1}(c-1)wt_i, & \text{if } k = 0 
\end{cases}$$
A Random Walk Acceleration

- **Algorithm Details:**
  - Phase II: Walk down **uniformly** (except the subtree that has been passed) until reaching a leaf node

\[
\begin{align*}
pu_0 &= 0.606 \\
pu_1 &= 0.564 \\
pu_2 &= 0.304 \\
pu_3 &= 0.075 \\
pu_4 &= 0
\end{align*}
\]
A Random Walk Acceleration

- **Algorithm Details:**
  - **Phase II:** Walk down *uniformly* (except the subtree that has been passed) until reaching a leaf node

```
\[ p_{u_0} = 0.606 \]
\[ p_{u_1} = 0.564 \]
\[ p_{u_2} = 0.304 \]
\[ p_{u_3} = 0.075 \]
\[ p_{u_4} = 0 \]
```

![Diagram of a random walk with probabilities and graph structure]

Each child node being chosen with probability \( \frac{1}{c} \)
A Random Walk Acceleration

● Algorithm Details:

● Phase II: Walk down uniformly (except the subtree that has been passed) until reaching a leaf node

\[
px_0 = 0, \\
px_1 = 0.075, \\
px_2 = 0.304, \\
px_3 = 0.564, \\
px_4 = 0.606
\]

Each child node being chosen with probability \( \frac{1}{c} \)
A Random Walk Acceleration

- **Algorithm Details:**
  - Phase II: Walk down uniformly (except the subtree that has been passed) until reaching a leaf node.

Each child node being chosen with probability $\frac{1}{c}$.
A Random Walk Acceleration

**Time Complexity:**

- **Phase I:** Walk up until obtain a tail from the coin (at level $k$) with its head probability
- **Phase II:** Walk down uniformly (except the subtree that has been passed) until reaching a leaf node

Each level is passed at most 2 times: $O(D)$

- $pu_4 = 0$
- $pu_3 = 0.075$
- $pu_2 = 0.304$
- $pu_1 = 0.564$
- $pu_0 = 0.606$
Experimental Settings

- Compared Algorithms:
  - TBF:
    Our tree-based framework + the random walk acceleration
  - Lap-GR:
    Laplacian Mechanism + The Greedy Algorithm
  - LAP-HG:
    Laplacian Mechanism + The HST-Greedy Algorithm
Experimental Settings

- **Datasets:**
  - **Synthetic datasets:** 200x200 Euclidean space

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\left</td>
<td>T \right</td>
</tr>
<tr>
<td>$\left</td>
<td>W \right</td>
</tr>
<tr>
<td>mean $\mu$</td>
<td>50, 75, <strong>100</strong>, 125, 150</td>
</tr>
<tr>
<td>standard deviation $\sigma$</td>
<td>10, 15, <strong>20</strong>, 25, 30</td>
</tr>
<tr>
<td>privacy budget $\epsilon$</td>
<td>0.2, 0.4, <strong>0.6</strong>, 0.8, 1</td>
</tr>
<tr>
<td>probability $p(T)$</td>
<td>$2 \times 10^4$, $4 \times 10^4$, $6 \times 10^4$, $8 \times 10^4$, $10 \times 10^4$</td>
</tr>
</tbody>
</table>

- **Real datasets:**
  - Trip records of passengers from Didi Chuxing

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>collected date</td>
<td>2016/11/01, ⋯, 2016/11/30</td>
</tr>
<tr>
<td>$\left</td>
<td>T \right</td>
</tr>
<tr>
<td>$\left</td>
<td>W \right</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>0.2, 0.4, <strong>0.6</strong>, 0.8, 1</td>
</tr>
</tbody>
</table>
Experimental Results

- Results on synthetic datasets

- Results on seal datasets
Outline

- Background
- Problem Definition
- A Tree-based Framework
- Random Walk Acceleration
- Experimental Evaluation
- Conclusions
Contributions

- Devise a novel tree-based framework for private online task assignment
- Design a privacy mechanism to protect location privacy
- Analyze the effectiveness of the framework
- Propose a random walk method for acceleration
Salamat! 谢谢 ありがとう お世話になりましてありがとうございます
Gracias ได้รับความสะดวก ขอขอบคุณ
Thank you