Elastic distinguishability metrics for Location Privacy

Marco Stronati
marco@stronati.org

joint work with
K. Chatzikokolakis and C. Palamidessi

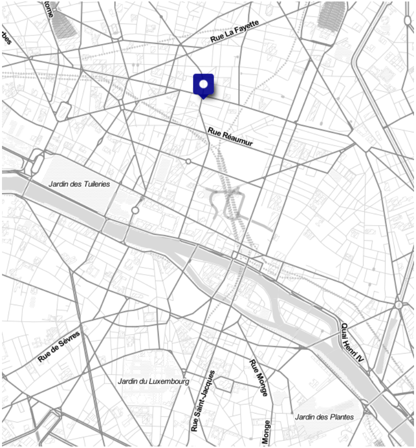
Privacy for LBS

- Goal: limit semantic inference
- (not anonymity)
- Reasonable utility for LBS

Obfuscation

Mechanism

$$x \longrightarrow \mathcal{M} \longrightarrow z$$

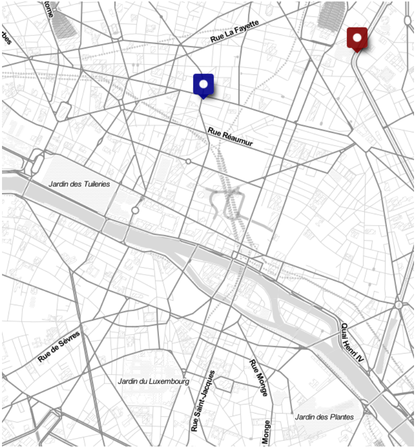


[Chatzikokolakis et. al: Broadening the Scope of Differential Privacy Using Metrics. PETS'13]

Obfuscation

Mechanism

$$x \longrightarrow \mathcal{M} \longrightarrow z$$

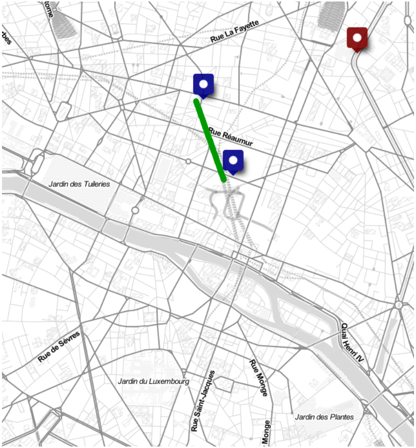


[Chatzikokolakis et. al: Broadening the Scope of Differential Privacy Using Metrics. PETS'13]

Obfuscation

Mechanism

$$x \longrightarrow \mathcal{M} \longrightarrow z$$

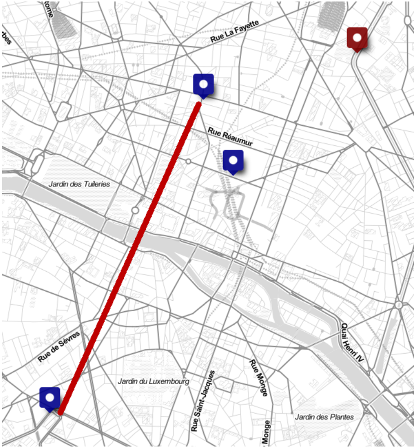


[Chatzikokolakis et. al: Broadening the Scope of Differential Privacy Using Metrics. PETS'13]

Obfuscation

Mechanism

$$x \longrightarrow \mathcal{M} \longrightarrow z$$



[Chatzikokolakis et. al: Broadening the Scope of Differential Privacy Using Metrics. PETS'13]

Obfuscation

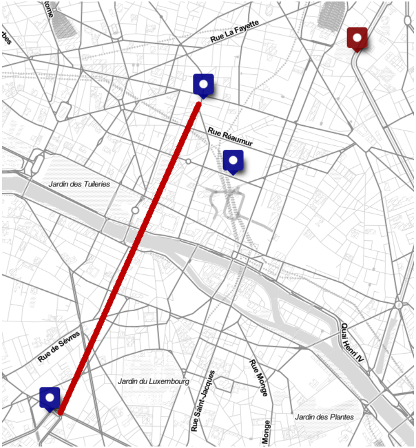
Mechanism

$$x \quad \longrightarrow \quad \mathcal{M} \quad \longrightarrow \quad z$$

$d_{\mathcal{X}}$-privacy

$$d_{\mathcal{P}}(\mathcal{M}(x), \mathcal{M}(x')) \leq d_{\mathcal{X}}(x, x') \quad \forall x, x'$$

- Distinguishability Metric on your set of secrets
- Apply noise according to the metric



[Chatzikokolakis et. al: Broadening the Scope of Differential Privacy Using Metrics. PETS'13]

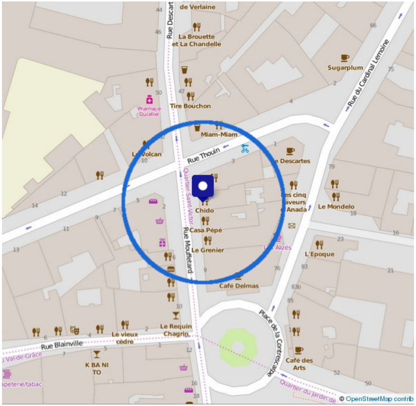
Geo Indistinguishability

$$d_{\mathcal{X}}(x, x') = \epsilon \, d_E(x, x')$$

- Space is privacy
- ϵ tunes how much

Requirement

I want to be indistinguishable from a certain amount of space.



[Andrés et al: Geo-indistinguishability: differential privacy for location-based systems. CCS'13]

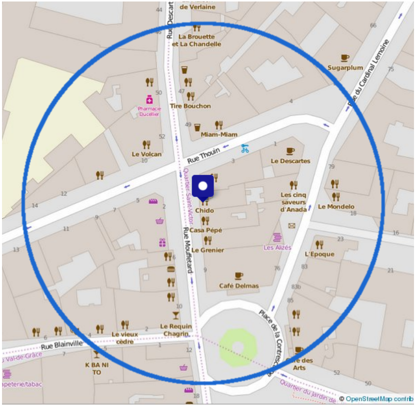
Geo Indistinguishability

$$d_{\mathcal{X}}(x, x') = \epsilon \, d_E(x, x')$$

- Space is privacy
- ϵ tunes how much

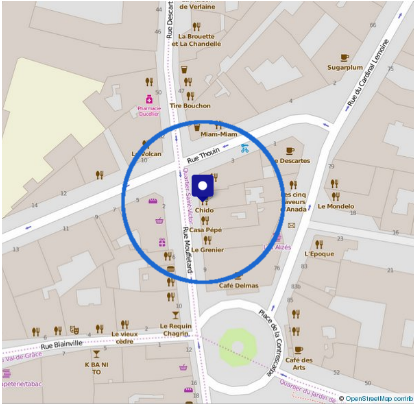
Requirement

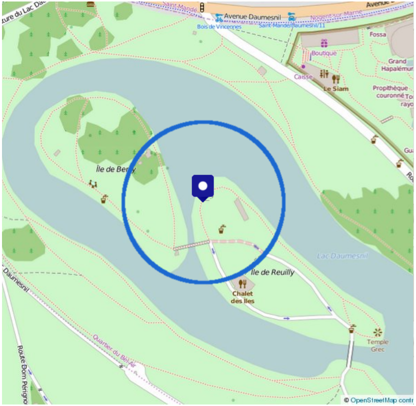
I want to be indistinguishable from a certain amount of space.



[Andrés et al: Geo-indistinguishability: differential privacy for location-based systems. CCS'13]

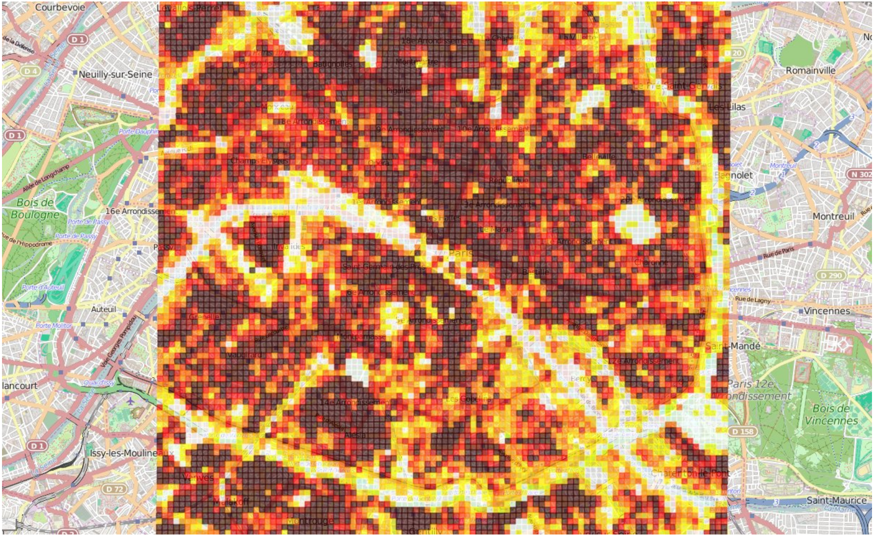
Not adaptable





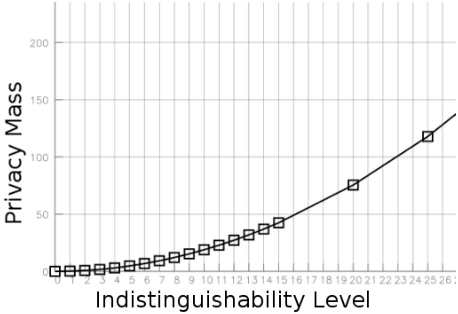
Privacy Mass from OpenStreetMap

Privacy Mass from OpenStreetMap



Privacy Requirement

I want to be indistinguishable from a certain amount of *privacy mass*.

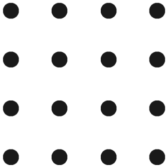


$$req(l) = mass$$

Building an Elastic Metric

Graph-based algo:

- start with a disconnetted graph
- interate over all nodes
  - compute `mass`
  - add an edge with $l = req^{-1}(\texttt{mass})$
- we stop at $l^{\top}$

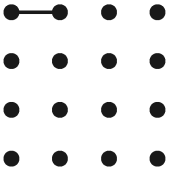


$$d_{\mathcal{X}}(x, x') = \texttt{shortest-path}(x, x')$$

Building an Elastic Metric

Graph-based algo:

- start with a disconnetted graph
- interate over all nodes
  - compute `mass`
  - add an edge with $l = req^{-1}(\texttt{mass})$
- we stop at $l^{\top}$

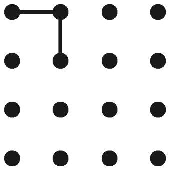


$$d_{\mathcal{X}}(x, x') = \texttt{shortest-path}(x, x')$$

Building an Elastic Metric

Graph-based algo:

- start with a disconnetted graph
- interate over all nodes
  - compute `mass`
  - add an edge with $l = req^{-1}(\mathtt{mass})$
- we stop at $l^{\top}$

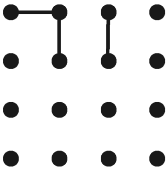


$$d_{\mathcal{X}}(x, x') = \mathtt{shortest\text{-}path}(x, x')$$

Building an Elastic Metric

Graph-based algo:

- start with a disconnetted graph
- interate over all nodes
  - compute `mass`
  - add an edge with $l = req^{-1}(\texttt{mass})$
- we stop at $l^{\top}$

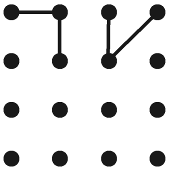


$$d_{\mathcal{X}}(x, x') = \texttt{shortest-path}(x, x')$$

Building an Elastic Metric

Graph-based algo:

- start with a disconnetted graph
- interate over all nodes
  - compute `mass`
  - add an edge with $l = req^{-1}(\texttt{mass})$
- we stop at $l^{\top}$

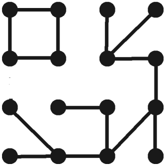


$$d_{\mathcal{X}}(x, x') = \texttt{shortest-path}(x, x')$$

Building an Elastic Metric

Graph-based algo:

- start with a disconnetted graph
- interate over all nodes
  - compute `mass`
  - add an edge with $l = req^{-1}(\texttt{mass})$
- we stop at $l^{\top}$

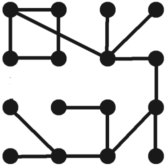


$$d_{\mathcal{X}}(x, x') = \texttt{shortest-path}(x, x')$$

Building an Elastic Metric

Graph-based algo:

- start with a disconnetted graph
- interate over all nodes
  - compute `mass`
  - add an edge with $l = req^{-1}(\texttt{mass})$
- we stop at $l^{\top}$



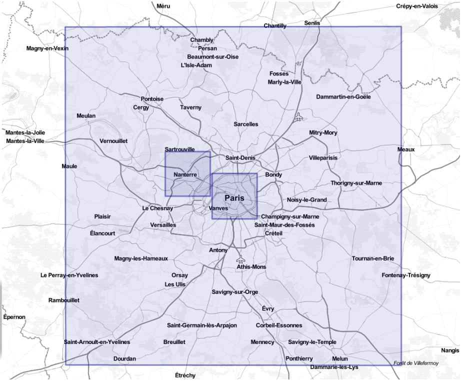
$$d_{\mathcal{X}}(x, x') = \texttt{shortest-path}(x, x')$$

Building an Elastic Metric

Graph-based algo:

- start with a disconnetted graph
- interate over all nodes
  - compute $\texttt{mass}$
  - add an edge with $l = req^{-1}(\texttt{mass})$
- we stop at $l^{\top}$

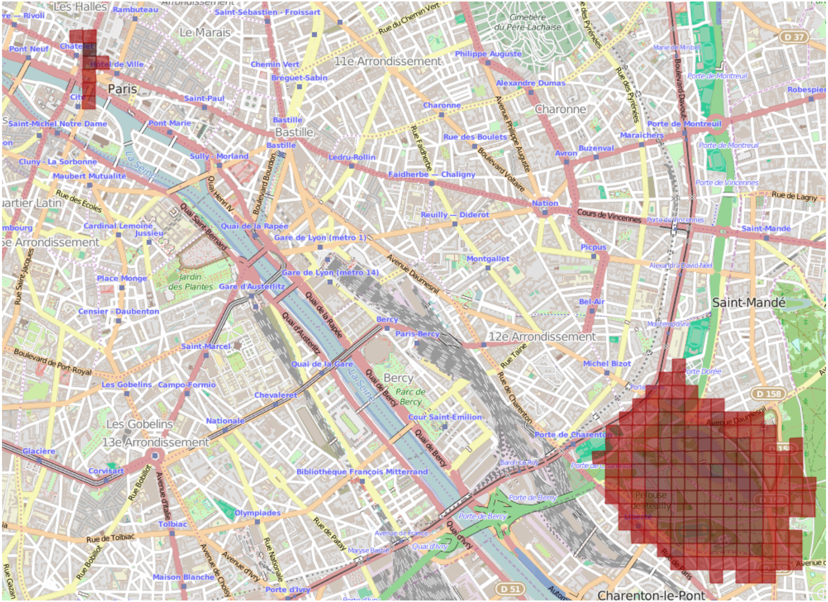
$$d_{\mathcal{X}}(x, x') = \texttt{shortest-path}(x, x')$$



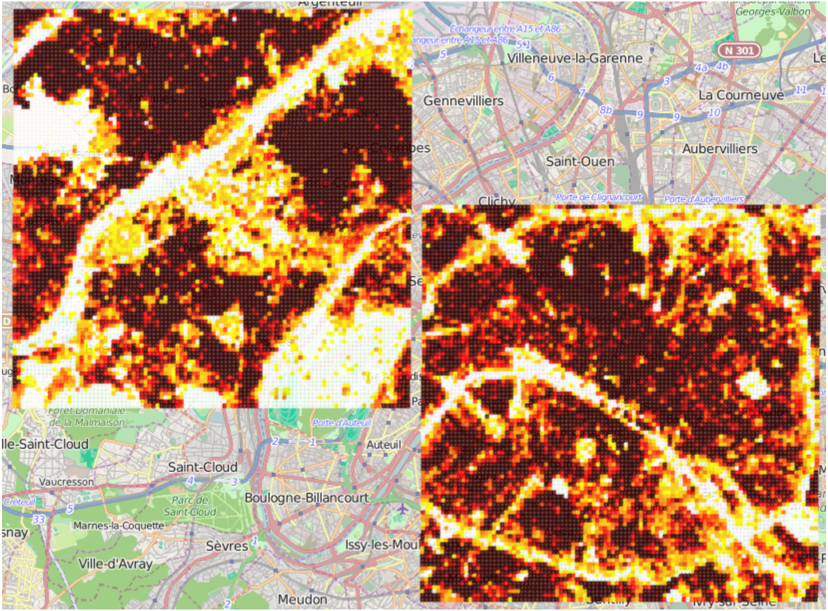
Elastic Mechanism

Elastic Mechanism = Elastic Metric + Exponential Mechanism

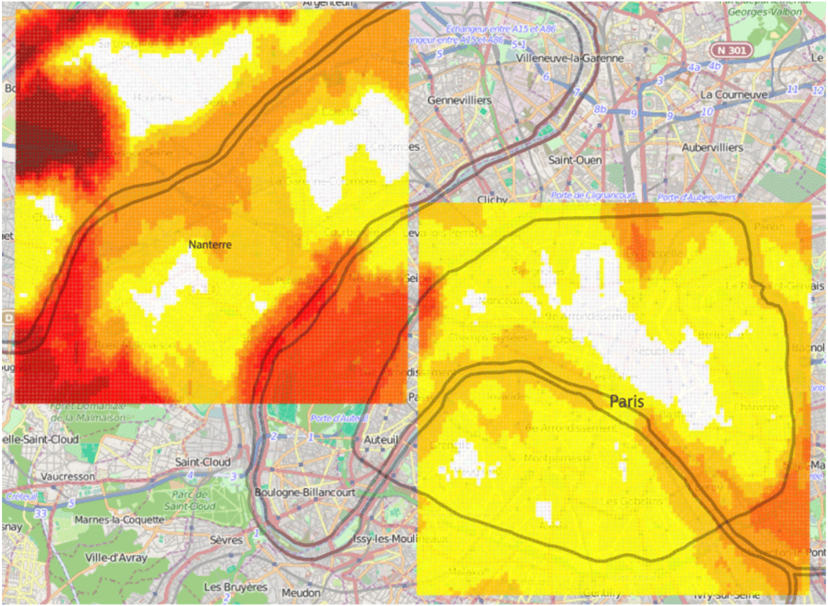
Elastic Mechanism



Elastic Mechanism

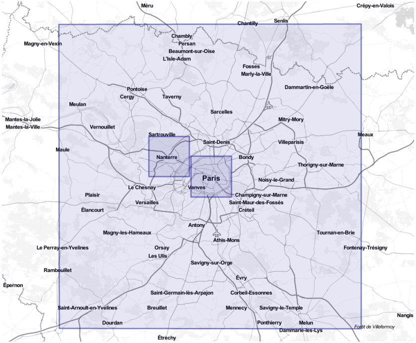


Elastic Mechanism



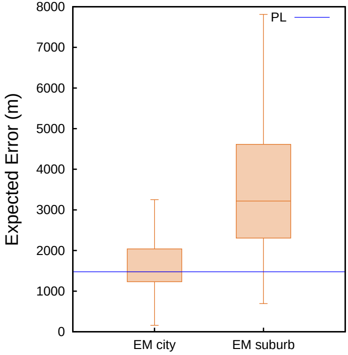
Evaluation

- EM vs PL
- City (Paris) vs Subsurb (Nanterre)
- Fixed Utility as Expected Error
- Compare Privacy as Adversarial Error
- Gowalla and Brightkite datasets

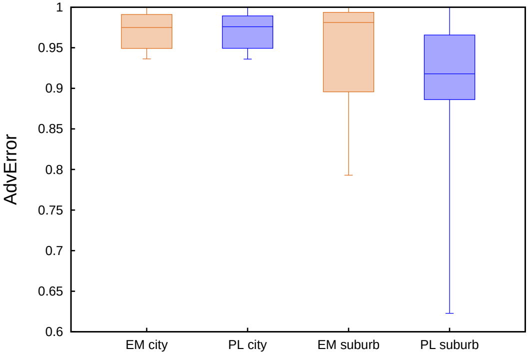


[Shokri, Theodorakopoulos, Boudec, Hubaux. Quantifying location privacy. S&P'11]

Evaluation



Expected Error (m)
8000
7000
6000
5000
4000
3000
2000
1000
0
PL
EM city
EM suburb



AdvError
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
EM city
PL city
EM suburb
PL suburb

Conclusion & Future

- Geoind is simple and efficient (Location Guard)
- Too rigid!

Contributions:

- Elastic metric with privacy mass requirement
- Scalable algorithm

Future Work:

- Include in privacy mass ideas from k-anonymity
- Lightweight version for Location Guard

Thanks

Don’t miss Location Guard tomorrow

Fences

- linear growth of epsilon
- fences for recurrent places
- achieve "better privacy" consuming less ϵ

$$d_F(x,x') = \begin{cases} d_{\mathcal{X}}(x,x') & x,x' \notin F \\ 0 & x,x' \in F \\ \infty & o.w. \end{cases}$$