Privacy by Projection: Federated Population Density Estimation by Projecting on Random Features

Zixiao Zong[1], Mengwei Yang[1], Justin Ley[1], Carter T. Butts[1,2,3] and Athina Markopoulou[1,3]

[1] Department of Electrical Engineering and Computer Science, [2] Departments of Sociology, Statistics, [3] Department of Computer Science, University of California-Irvine

Motivation

- Consider population density estimation based on location data crowdsourced from mobile devices, using kernel density estimation;
- Uploading users' locations to server raises privacy concerns;

Goal: Propose a Federated KDE framework for estimating the user population density, which not only keeps users' data local, but also protects against a malicious server that tries to infer users' locations.

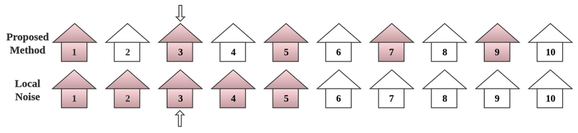
Introduction

Objective

- Population density is estimated on a grid with a chosen range, where density at each coordinate is estimated from users' locations;
- Perform in a distributed manner, in which users do not directly share their locations with the server, and such information cannot be inferred by a malicious server.

Compare to local noise-adding scheme

- Delocalized Projection vs. Localized Noise.



Preliminaries

Notation and Problem Setting

- N users, and each user i is associated with a location $\mathbf{d}_i$ ($\mathbf{d}_i \in \mathbb{R}^2$);
- Full dataset $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$;
- Consider estimating density on a $P \times Q$ grid, with $G(\mathbf{g}_{pq})$ being the density obtained at location (p, q) estimated over all N data points.

Kernel Density Estimation

- Kernel Density Estimation (KDE) is a non-parametric method to estimate a density function from a set of random draws from the corresponding distribution.

$$f(\mathbf{x}|\mathbf{D}) = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}|\mathbf{d}_i) = \frac{1}{N}\sum_{i=1}^{N} k_h(\mathbf{x}, \mathbf{d}_i) \tag{1}$$

- Use gaussian kernel $k_h(x, y) \equiv \exp(-\frac{\Delta^2}{2h^2})$, $\Delta = \|x - y\|_2^2$ throughout;

Random Fourier Feature (RFF)

- Shift-invariant kernel follows Bochner's theorem

$$k(\mathbf{x}, \mathbf{y}) = \int p(\omega) e^{j\omega^\top(\mathbf{x}-\mathbf{y})} d\omega = \mathbb{E}\left[e^{j\omega^\top(\mathbf{x}-\mathbf{y})}\right] \tag{2}$$

- (2) can be approximated by Monte-Carlo sampling

$$\mathbb{E}\left[e^{j\omega^\top(\mathbf{x}-\mathbf{y})}\right] \approx \frac{1}{B}\sum_{b=1}^{B} e^{j\omega_b^\top(\mathbf{x}-\mathbf{y})} = \frac{1}{B}\sum_{b=1}^{B} \phi_b(\mathbf{x}, \mathbf{y}) \tag{3}$$

- where ϕ_b is basis function

$$\phi_b(\mathbf{x}, \mathbf{y}) = [\cos(\omega_b^\top \mathbf{x}), \sin(\omega_b^\top \mathbf{x})][\cos(\omega_b^\top \mathbf{y}), \sin(\omega_b^\top \mathbf{y})]^\top \tag{4}$$

Methods

Baseline: Federated KDE

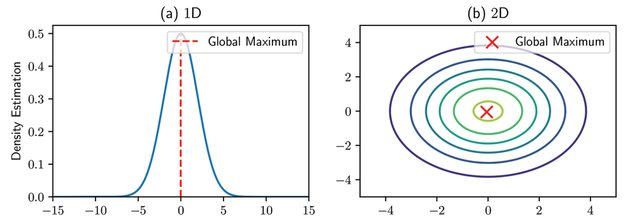
- From (1), KDE is linearly separable, which lands itself to federated setting: each user i is responsible for calculating $f(\mathbf{g}_{pq}|\mathbf{d}_i)$;
- Formally speaking, a user i will evaluate $G_h(\mathbf{d}_i)$ along the $P \times Q$ grid with bandwidth h, producing

$$G_h(\mathbf{d}_i) = \begin{pmatrix} k_h(\mathbf{d}_i, \mathbf{g}_{11}) & \cdots & k_h(\mathbf{d}_i, \mathbf{g}_{1Q}) \\ \vdots & \ddots & \vdots \\ k_h(\mathbf{d}_i, \mathbf{g}_{P1}) & \cdots & k_h(\mathbf{d}_i, \mathbf{g}_{PQ}) \end{pmatrix} \tag{5}$$

- Overall density surface can be estimated by aggregating $\frac{1}{N}\sum_{i=1}^{N} G_h(\mathbf{d}_i) \in \mathbb{R}^{P\times Q}$.

Federated KDE privacy attack

- Federated KDE fully reveals users' privacy.



Proposed Algorithm: Federated RFF KDE

- Substitute (3) into (1), and then get

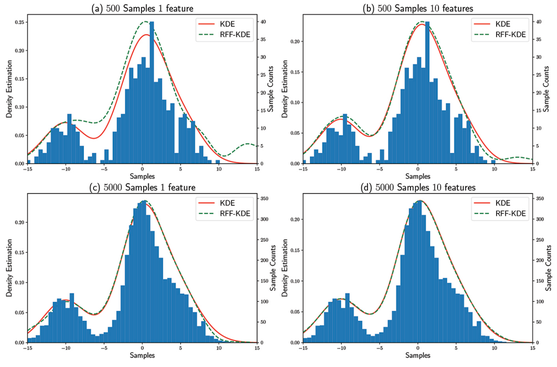
$$f(\mathbf{x}|\mathbf{D}) \approx f'(\mathbf{x}|\mathbf{D}) = \frac{1}{NB}\sum_{i=1}^{N}\sum_{b=1}^{B} \phi_b^i(\mathbf{x}, \mathbf{d}_i) \tag{6}$$

- RFF is used to approximate kernel function;
- Each user generates s/he own basis ω_b^i without sharing them with anyone else;
- Each user's work is calculating

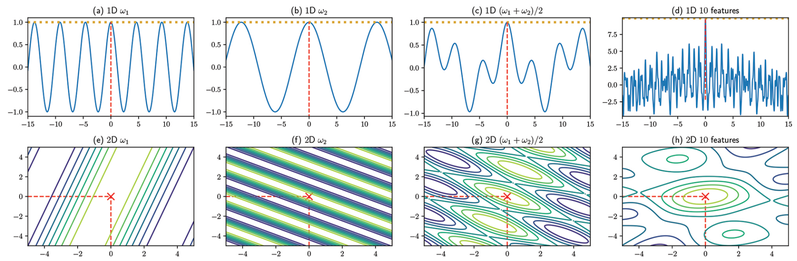
$$G'(\mathbf{d}_i) = \begin{pmatrix} \sum_{b=1}^{B} \phi_b^i(\mathbf{d}_i, \mathbf{g}_{11}) & \cdots & \sum_{b=1}^{B} \phi_b^i(\mathbf{d}_i, \mathbf{g}_{1Q}) \\ \vdots & \ddots & \vdots \\ \sum_{b=1}^{B} \phi_b^i(\mathbf{d}_i, \mathbf{g}_{P1}) & \cdots & \sum_{b=1}^{B} \phi_b^i(\mathbf{d}_i, \mathbf{g}_{PQ}) \end{pmatrix} \tag{7}$$

- Density surface is estimated with $\frac{1}{NB}\sum_{i=1}^{N} G'(\mathbf{d}_i) \in \mathbb{R}^{P\times Q}$.

Convergence Analysis



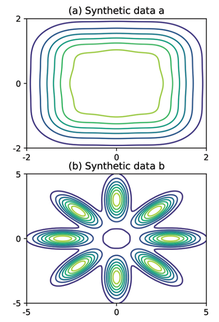
Privacy Analysis

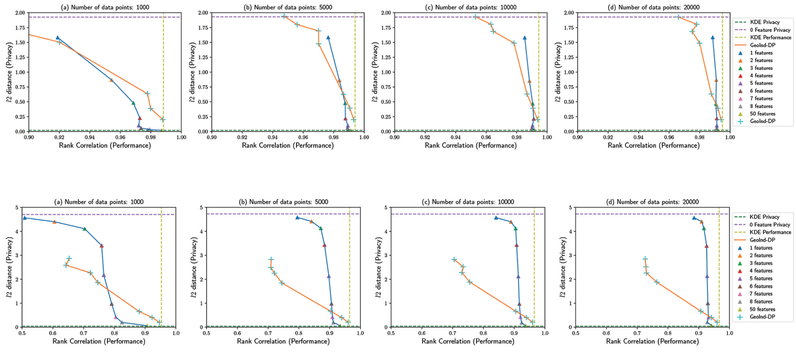


- A local maximum MUST appear at user's location, and server cannot distinguish it from other local maxima;
- More features lead to sparse local maxima patterns, so less features are encouraged to use whenever possible;
- As long as a user always uses the same set of basis, no matter how many queries are sent by server (even unnecessary), the server still cannot figure out user's location (even user's moving).

Numerical Tests

- Performance Measurement: Rank Correlation;
- Privacy Measurement: $L2$ distance based metric, which represents the distance from the best inferred position to the real location;
- Baseline 1: Federated KDE (Best performance and Worst privacy);
- Baseline 2: 0 features, where no information is shared with server, so server cannot do better than random guessing (Best privacy);
- Compared with Geo-Indistinguishability.





Publication

PETs(2023): This paper is to appear in Proceedings on Privacy Enhancing Technologies (PoPETs), July 10–14, 2023, Lausanne, Switzerland.

This research has been supported by NSF Awards 1900654, 1956393, 1939237, NIH 1R01GM144964-01, and by a UCI Seed grant from the Office of Research.