Real-World Trajectory Sharing with Local Differential Privacy

We focus on the challenging problem of sharing location sequence data privately. Although trajectory data is incredibly useful for researchers and analysts, existing mechanisms to share such data privately have a number of limitations. Our local differentially private solution uses a distance-based mechanism to perturb overlapping n-grams (i.e., contiguous subsequences of length $n$) of trajectory data. We utilize a multi-dimensional hierarchy over publicly available external knowledge of places of interest to improve the realism and utility of the shared trajectories. Our experiments, using real-world data and a range of queries, each with real-world application analogues, demonstrate the superiority of our approach over a range of alternative methods. Importantly, we show that including real-world public data does not negatively affect privacy or efficiency.

Introduction

Sharing trajectories is beneficial for many real-world applications, such as managing disease spread through contact tracing and tailoring public services to a population’s travel patterns. However, public concern over privacy and data protection has limited the extent to which this data is shared in ways that are useful for end-users.

Local differential privacy (LDP) offers a framework that enables data sharing in which users share a perturbed version of their data in a way that a level of privacy can be guaranteed. Although trajectory data has been synthesized and published in the centralized DP domain (e.g., mechanisms in the local setting are limited. Moreover, existing LDP mechanisms typically assume equal sensitivity between outputs and/or fail to incorporate user-independent public knowledge, which makes them too restrictive, gives unrealistic outputs, and ultimately leads to low practical utility.

Problem

We model trajectories as the time-ordered sequence of the places of interest (POIs) visited by someone in one day, as illustrated in Figure 1. We want each user to perturb their own trajectory, which is then shared with the aggregator – this provides a stronger level of privacy. Specifically, we provide a privacy guarantee through the use of e-LDP, which is stricter than existing distance-based mechanisms (e.g., that rely on LDP relaxations.

To ensure realism and boost utility, we also impose a ‘reachability’ constraint that prevents illogical outputs (e.g., traveling from London to Copenhagen in twenty minutes).

Using External Knowledge

To boost utility further, we utilize publicly available external knowledge (e.g., business locations and opening times, public transport schedules) to make certain outcomes more likely. We incorporate this by using a semantic distance function to influence the probability distribution function of the exponential mechanism. This ensures more semantically similar outputs are more likely to be returned. For example, if Jane is at a pizza restaurant, she is more likely to be perturbed to a Greek restaurant than a bowling alley. Importantly, as using this information is independent of the real data, no privacy budget needs to be spent.

n-gram Solution

Our n-gram solution comprises three main parts: hierarchical decomposition, n-gram perturbation, and trajectory reconstruction. Figure 2 shows an overview of the mechanism.

Hierarchical Decomposition

To ensure our mechanism remains efficient and scalable, we project our space into a finite number of regions. We exploit the hierarchical structure that is inherent in the space, time, and category data to create coarse space-time-category (STC) regions. An example STC region is: (NW of town, 9-10pm, Restaurant).

Once hierarchical decomposition is complete, we pre-process our input trajectories to convert them from the POI level to the STC region level.

Perturbation

We perturb overlapping n-grams using our semantic distance function and the exponential mechanism. Using the exponential mechanism, and exploiting the composition property of LDP, ensures that the privacy loss is no more than the specified privacy budget, $\varepsilon$.

The perturbation process is illustrated in Figure 3. As the ends of each trajectory are not perturbed $n$ times, we perform supplementary perturbations with shorter n-grams, following the same mechanism as the pre-main set of perturbations.

Trajectory Reconstruction

Perturbation means that each point in the trajectory has a points associated with it. To rectify this, we use optimization to identify a feasible trajectory that minimizes the error between the perturbed n-grams and the output trajectory. As this only involves the perturbed data, we invoke LDPs post-processing property, which means that no additional privacy budget needs to be used. Finally, we generate trajectories on the POI level by randomly sampling POIs within each STC region of the reconstructed trajectory.

Evaluation

We evaluate our mechanism using two datasets from New York City: taxi trajectories, and SafeGraph (SG) check-in sequences. We compare our method (NGram) with four baselines:

- IndReach: perturbs POI-timestep pairs independently
- IndNoReach: IndReach, but no reachability constraint
- PhysDist: NGram, but with no external knowledge
- NGramNoH: NGram, but without STC regions

Table 1 shows the normalized semantic distance between the real trajectories and the perturbed trajectories. It also shows the average runtime for each mechanism. NGram is the best performing mechanism when considering popularity and efficiency.

Future Work

Future work includes focusing on expanding the range of external information used to include aspects such as time-varying popularity and less-structured data (e.g., semantic information from tweets), as well as offering a framework for personalized LDP. Our work can also be extended to other forms of trajectories, such as transaction patterns or internet browsing history.

References