.

Preserving Differential Privacy in Publication of Trajectory Data

Kulkarni Sourabh Shrinivasrao, 21114053

Abstract— **For development and improvement of many modern day applications, collecting and publishing the trajectory data is very important. But publishing it right away possesses different challenges such as compromise in user's privacy. Also, traditional trajectory merging algorithms are too slow. We try to solve both of these problems using our database publication algorithm and trajectory merging algorithm. Theoretical analysis shows that our privacy loss is much less.**

Index Terms—

1. *Differential Privacy*
2. *Trajectory Data*
3. *Database Publication*
4. *Laplace Noise*

INTRODUCTION

With the revolution in the devices able to accommodate the GPS services , generating and collecting the trajectory data is becoming increasingly easier for users of those devices. Of course, This data is very much useful for inventors, researchers and solving real life problems. But this also comes with a cost. Publishing the trajectory data as it is exposes sensitive details of users such as their routines, habits, personal information etc.

Differential privacy is an elegant and modern approach to protect each individuals privacy while using their data in the database. Third party with access to the database cannot reach a conclusion whether the data belongs to an user or not. Using Laplace mechanism is the most commonly used method to achieve differential privacy adding noise in the **true count** (trajectory count on a road). While adding this noise, one must look after the boundedness of the noise else making release meaningless. Many work is available on this topic such as first applied differential privacy to sequential data, n-gram model etc.

TERMS AND DEFINITIONS

1.Differential Privacy: If the output of a query does not depend on the presence of a single record, then we can say this database posseses differential privacy.

2.Sensitivity: for a function f: D $\rightarrow R^d$, the sensitivity df = $_{max}||f(D1)-f(D2)||$ where D1, D2 differ in atmost 1 record.

3.Laplace mechanism: pdf(x)=(1/2b) $e^{-|x-m|/b}$ where m is mean and b=df/E where E is privacy budget.

4.Trajectory: a list of location-time pairs: T = $(l_1,t_1) \rightarrow (l_2,t_2) \rightarrow \ldots \rightarrow (l_{|T|},t_{|T|})$

5.True count: If a trajectory exists in the database, True count=1. Else 0.

CLAIM:

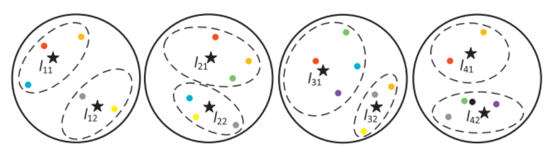
Unbounded Laplace noise does not yield preservation of utility and privacy. Thus, we will use bounded noises.

PROCEDURE:

Original Database

No.	t_1	t_2	t_3	t_4	Trajectry
T_1	1	1	1	1	(l_1, t_1)->(l_2, t_2)->(l_3, t_3)->(l_4, t_4)
T_2	1	1	1	1	(l_5, t_1)->(l_6, t_2)->(l_7, t_3)->(l_8, t_4)
T_3	1	1	1	0	(l_9, t_1)->(l_{10}, t_2)->(l_{11}, t_3)
T_4	0	1	1	1	(l_{12}, t_2)->(l_{13}, t_3)->(l_{14}, t_4)
T_5	1	1	1	1	(l_{15}, t_1)->(l_{16}, t_2)->(l_{17}, t_3)->(l_{18}, t_4)
T_6	0	0	1	1	(l_{19}, t_3)->(l_{20}, t_4)
T_7	1	1	1	0	(l_{21}, t_1)->(l_{22}, t_2)->(l_{23}, t_3)

Step One. Location Generalization



Step Two. Trajectory Data Releasing

Original Tr.	Generalized Tr.	Real Count	Noisy Count
T_1	l_{11}->l_{21}->l_{31}->l_{41}	1	2.46781
T_2	l_{11}->l_{21}->l_{32}->l_{41}	1	1.83188
NULL	l_{11}->l_{22}->l_{31}->l_{42}	0	0.705873
T_3	l_{11}->l_{22}->l_{31}	1	1.58152
T_4, T_6	l_{21}->l_{31}->l_{42}	2	2.71865
NULL	l_{11}->l_{21}->l_{31}->l_{42}	0	1.92746
T_5, T_7	l_{12}->l_{22}->l_{31}->l_{42}	2	2.82741
NULL	l_{12}->l_{22}->l_{32}->l_{42}	0	1.8422

Pictorial representation of general idea

The general idea is, we want to add some "fake" (generated) trajectories in the database to the original trajectories. While doing this we will not include all of the original trajectories in order to match the original size of the database. After that, we will add noise to the true counts in order to ensure that these trajectories become indistinguishable. We can achieve this by following these steps:

Step1: curating the Dataset.

The author state that they have obtained trajectory data from Microsoft Research's T-Drive project with trajectories of 10000+ taxis and more than 15 million datapoints. They took about 850 trajectories out of it. But of course this needs very high computational power so we limit that number to about 200 trajectories. Also, our original dataset contains txt files with each file containing one trajectory. We update that into merging all trajectories into 1 single txt file for ease of handling. Originally, each file in the dataset had entries of the format:

```
{ taxi id, date time, longitude, latitude }
```

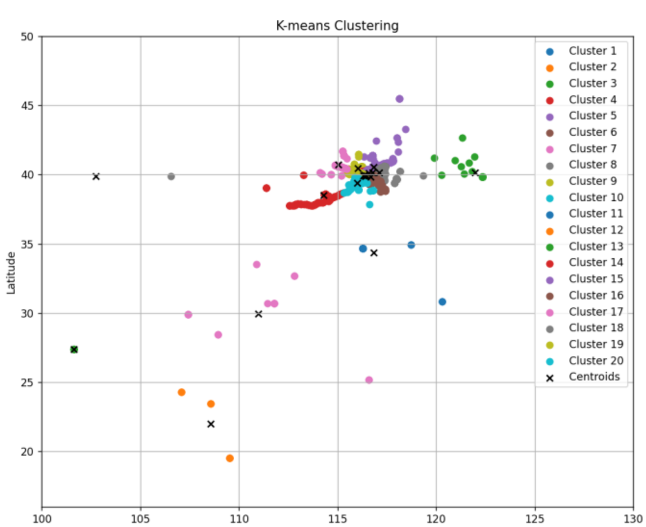
```
dataset > 5.txt
1   5,2008-02-02 13:43:04,116.62934,39.82726
2   5,2008-02-02 13:53:04,116.62934,39.82725
3   5,2008-02-02 14:03:04,116.62933,39.82725
4   5,2008-02-02 14:13:04,116.61582,39.82817
5   5,2008-02-02 14:23:08,116.59818,39.82816
6   5,2008-02-02 14:33:04,116.62944,39.82734
7   5,2008-02-02 14:43:04,116.62946,39.82743
8   5,2008-02-02 14:53:04,116.62945,39.82746
```

a sample file in the initial dataset

After curation, we create a single file with true counts appended at the end of each entry(which are 1 initially).

Step2: Clustering the Location universe into K-clusters.
After curating the dataset, we cluster it into K different clusters using standard K-means clustering algorithm. Although the paper suggests a different clustering algorithm which may be possible theoretically, it is not made clear nor is inferable from paper how one would actually implement that in the code.

After applying clustering, we get following partitions(K=20)



Step3 : Generation of General trajectories.
In this step. From the clusters, we generate n1 trajectories (n1 is of user's choice. Here we take n1=50). We do this by keeping length of each new trajectory at least 25 points and we randomly select a cluster and a random point in the cluster with a timestamp generator. Of course, true counts will be 0 and to be appended at end of each record.

Step4: generalizing all of the trajectories.
Based on the clusters, we replace each entry's longitude and latitude by the number of cluster that location is. As a consequence , there might be duplicate trajectories. In that case, we increase the true count and only keep unique trajectories.

Step5: Adding bounded Laplace noise in true counts.

To ensure differential privacy, we add bounded Laplace noises in the true counts with following algorithm:

Input: $\alpha = 0, \beta, \{tc_i | i = 1, \ldots, n_1\}, i = 0$
Output: $\{nc_i | i = 1, \ldots, N\}$ or $\perp$

01. **If** $\alpha < 0 \,\|\|\, \alpha < \beta \,\|\, n_1 \leqslant 0$, **then**
02. **Return** $\perp$
03. **For all** $p, q \in [1, n_1]$
04. $\mu = ComputeAverage\{tc_p\}$, $\Delta f = \max_{p,q}\{|tc_p - tc_q|\}$
05. $b = \Delta f / \mu$, $\beta = 2 * \mu$
06. $ln_i \leftarrow pdf(x)$
07. **While** $ln_i < \beta$ && $ln_i > \alpha$ && $i < n_1$
08. **Do** $i++$, $ln_i \leftarrow pdf(x)$, $nc_i \leftarrow tc_i + ln_i$
09. **If** $i \neq n_1$, goto line 06
10. $ln_i \leftarrow pdf(x)$
11. **While** $ln_i < \beta$ && $ln_i > \alpha$ && $i < n$
12. **Do** $i++$, $n_i \leftarrow pdf(x)$, $nc_i \leftarrow ln_i$
13. **If** $i \neq n$, goto line 10
14. **Return** $\{nc_i | i = 1, \ldots, N\}$

This algorithm basically generates a random Laplace noise and checks before adding if it fits in the bounds defined by alpha and beta.

As a result of all these steps, our differentially private dataset looks like this:

```
Trajectory: 9,9,0,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,17,17,17,17,17,
Trajectory: 17,17,17,9,9,9,9,9,9,9,9,9,9,9,17,17,17,17,17,17,17,17,
Trajectory: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Trajectory: 17,17,17,17,17,17,17,17,9,9,17,17,17,9,11,11,11,11,11,1
Trajectory: 17,17,17,17,2,2,8,8,4, Count: 2.1722764883727512
Trajectory: 17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,1
Trajectory: 9,9,9,9,9,9,9,9,9,9,9,9,17,17,17,17,0,0,0,9,0,0,0,0,0,1
Trajectory: 17,9,6,9,8 8,7, Count: 1.2018920448093702
Trajectory: 0,0,9,9,9,17,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,9,9,9,9,9,9,
Trajectory: 2,2,2,8,9, Count: 1.0084984312980698
Trajectory: 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,17,17,17,17,17,17,17,
Trajectory: 0,2,2,2,0,17,0,0,9,1, Count: 1.9303985918945799
Trajectory: 17,9,9,9,9,9,9,9,17,17,17,9,9,9,17,17,17,17,17,17,17,17
Trajectory: 17,17,17,17,17,17,17,17,17,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
Trajectory: 11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,1
Trajectory: 9,9,9,9,9,9,9,9,9,9,9,17,17,9,9,9,9,9,9,9,9,9,9,9,9,0,0
```

Released Database

.

Proof that differential privacy is preserved

For any sequence r of outcomes $r_i \in Range(A_i)$, i = 1, 2, we write $A^r{}_1$ and $A^r{}_2$ for algorithm A1 and A2 supplied with r1 and r2. The probability of output r from the sequence $A^r{}_1$ (D) and $A^r{}_2$ (D) on database D is

$$Pr[A(D) = r] = Pr[A_1^r(D) = r_1]Pr[A_2^r(D) = r_2].$$

Applying definition of differential privacy, we get:

$$\begin{aligned} Pr[A_1^r(D) = r_1]Pr[A_2^r(D) = r_2] &\leq (e^{\epsilon_1}Pr[A_1^r(D') = r_1]) * (e^{|T|\cdot\epsilon_2}Pr[A_2^r(D') = r_2]) \\ &= e^{\epsilon_1+|T|\cdot\epsilon_2}Pr[A_1^r(D') = r_1]Pr[A_2^r(D') = r_2] \\ &= e^{\epsilon_1+|T|\cdot\epsilon_2}Pr[A_1^r(D') = r_1 \wedge A_2^r(D') = r_2] \\ &= e^{\epsilon_1+|T|\cdot\epsilon_2}Pr[A(D') = r] \end{aligned}$$

hence, we can achieve differential privacy.

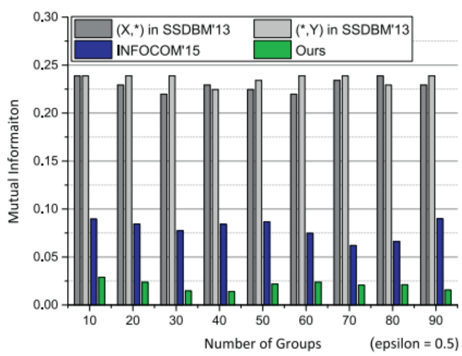
Results

Lets Define Mutual information metric as following:

$$MI(x,y) = \sum_t \sum_{x(t)} \sum_{y(t)} Pr(x(t),y(t)) \log \frac{Pr(x(t),y(t))}{Pr(x(t))Pr(y(t))}$$

Which calculates the dependency between two variables X and Y. If X and Y are not dependent, we can know nothing about the second if we know about first and vice versa.

Using this metric, we compare our algorithm of data publication with other as shown in below bar diagram. Clearly the less MI, the better algorithm.



Conclusion

In this paper, we saw how it is inappropriate to release trajectory data without any treatment. With given algorithms, one can achieve differential privacy with first applying clustering algorithm and then by adding bounded Laplace noises.

.

References

[1] "*Achieving differential privacy of trajectory data publishing in participatory sensing*" by Meng Li, Liehuang Zhu, Zijian Zhang , Rixin Xu. [Link to the paper](#).

[2] Source for many images is from the implementation of this paper by me: [GitHub link for code](#)

[3] [What is differential privacy](#)