International Journal of Information Security (2021) 20:511–533
https://doi.org/10.1007/s10207-020-00516-5

REGULAR CONTRIBUTION

Enhancing spatial and temporal utilities in differentially private moving objects database release

Fatemeh Deldar[1] · Mahdi Abadi[1]

Published online: 24 July 2020

Abstract

The pervasive use of mobile technologies and GPS-equipped vehicles has resulted in a large number of moving objects databases. Privacy protection is one of the most significant challenges related to moving objects databases because of the legal requirements in many application domains. Over the last few years, several differentially private mechanisms have been proposed for moving objects databases. However, most of them aim to answer statistical queries and do not release a differentially private version of a moving objects database. In this paper, we present DP-MODR, a differentially private (DP) mechanism for synthetic moving objects database release (MODR). DP-MODR tries to efficiently and effectively release synthetic trajectories while preserving spatial and temporal utilities. In this way, the released differentially private moving objects database can be used for different purposes as well, including data analysis tasks. DP-MODR keeps some main spatial and temporal properties of original trajectories and defines a new differentially private tree structure to keep the most probable paths with different lengths and different starting points, which are then iteratively joined to generate synthetic trajectories in a bottom-up way. Also, we present an extension of DP-MODR to support moving objects databases whose locations are time-dependent. Extensive experiments on real moving objects datasets using multiple spatial and temporal evaluation measures show that DP-MODR enhances the utility of query answers and better preserves the main spatial and temporal properties of original trajectories in comparison with recent related work.

Keywords Differential privacy · Moving objects database release · Noisy cost-sensitive path tree · Spatial utility · Temporal utility · Time-dependent query

1 Introduction

The popularity of location-based services and applications is growing with the rapid growth of smartphone owners, resulting in the rapid growth of moving objects databases. A moving objects database is a multiset of trajectories, each of which represents the movement history of a moving object during a period of time. Moving objects databases offer a vast application potential for researchers and enterprises, and there is a great interest in mining these databases for purposes such as city planning, traffic control, trajectory pattern analysis, and municipal transportation. For example, transport authorities can use moving objects databases for better designing transportation systems and optimizing resource consumption. However, unauthorized exposure of moving objects' trajectories may disclose their trip histories, home and work locations, frequent meeting points, or visits to sensitive locations such as hospitals, health clinics, and airports. The disclosure of such information has always been of concern to the owners of trajectories and prevents them from sharing their trajectories in moving objects databases.

Traditional privacy protection techniques for moving objects databases have mostly focused on location privacy, which is often achieved by perturbing or obfuscating each point of a trajectory. However, these location-based techniques are not usually sufficient for protecting the spatial and temporal properties of trajectories. On the other hand, anonymized moving objects databases that do not contain personal identifiers or other evidence of identity still do not prevent the precise identification of moving objects [13]. For example, it was shown that 87% of the population in the USA had reported characteristics that likely made them be

✉ Mahdi Abadi
abadi@modares.ac.ir

[1] School of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

uniquely identified, even though all explicit identifiers were removed from data records [32].

Differential privacy [9,40] has emerged as one of the strongest privacy definitions for privacy protection. The intuition is that the same conclusions must be reached independently of whether an individual data record opts into or opts out of a database. Specifically, it ensures that the probability that a statistical query will produce a given result is approximately the same as when one data record is added or removed from a database. Differential privacy provides strong privacy guarantees independently of the background knowledge of the adversary [12,19]. This is because differential privacy is a property of the data release mechanism, not of an interaction between the mechanism and the adversary [10]. Thus, differentially private mechanisms are immune to a wide range of privacy attacks [12]. Initially, work on differential privacy mainly concentrated on answering statistical queries [5,7,9,28]. However, some recent work has begun to use differential privacy for data release scenarios in different fields [1,29,39].

In the last years, several differentially private mechanisms have been proposed to answer statistical queries over moving objects databases [3,8,17,33]. However, as mentioned, the majority of them do not release a differentially private (synthetic) version of an original moving objects database. Although some few mechanisms have been proposed to address this issue [15,16], they cannot properly preserve the spatial and temporal properties of original trajectories. In this paper, we continue this line of research by presenting DP-MODR, a differentially private mechanism for synthetic moving objects database release that preserves spatial and temporal utilities as much as possible. In this mechanism, we first derive some useful properties of an original moving objects database, including number of trajectories, number of points in each trajectory, and mobility patterns of trajectories, in a differentially private way. Then, we construct some so-called noisy cost-sensitive path trees to keep existing most probable paths with different lengths (up to a maximum length) and different starting points. Finally, using these noisy cost-sensitive path trees and by considering the obtained differentially private spatial and temporal properties of original trajectories, we efficiently construct a synthetic moving objects database. Furthermore, we extend DP-MODR to support moving objects databases whose locations are time-dependent. In this new extension, also known as DP-MODRT, the synthetic moving objects database can preserve the time information of trajectories as well as the location information, in a differentially private way.

In the following, we list the main contributions of this paper:

- We introduce DP-MODR, a differentially private mechanism for synthetic moving objects database release, which aims to enhance both spatial and temporal utilities simultaneously. DP-MODR achieves this aim by preserving the spatial and temporal properties of original trajectories in synthetic trajectories, in a differentially private manner.
- We present a new tree structure, known as a noisy cost-sensitive path tree, to keep existing most probable paths with different lengths and different starting points while satisfying differential privacy. We efficiently use the noisy cost-sensitive path trees to generate synthetic trajectories.
- We efficiently construct a differentially private moving objects database by generating synthetic trajectories in a bottom-up way. Each synthetic trajectory is generated by iteratively joining the most probable paths until the intended length of that trajectory is reached.
- We design an attack, called sensitive locations disclosure attack, on synthetic moving objects databases and show to what extent DP-MODR is resilient to it.
- We extend DP-MODR to support moving objects databases whose locations are time-dependent. The new differentially private mechanism, also known as DP-MODRT, is especially suitable for answering time-dependent queries over a synthetic moving objects database.
- Through extensive experiments on real moving objects datasets, we show that DP-MODR enhances the utility of query answers and better preserves the main spatial and temporal properties of original trajectories in comparison with recent related work. Also, through some experiments, we show that DP-MODRT can preserve the time information of trajectories as well as the location information.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 provides some preliminaries and basic definitions. Section 4 introduces DP-MODR, explains it in detail and analyzes its privacy guarantee and performance. In Sect. 5, we extend DP-MODR to support moving objects databases whose locations are time-dependent. In Sect. 6, we report our experimental results in detail, and finally, in Sect. 7, we give a summary and discussion.

2 Related work

In this section, we review the state-of-the-art mechanisms for preserving differential privacy in moving objects databases.

The notion of differential privacy was introduced by Dwork [9] in 2006, and since then, it has been successfully applied to a wide range of data analysis tasks [4,6,18,35,37]. To maximize the utility of the results provided by differential privacy, the magnitude of the random noise should be as small as possible. The basic idea is to concentrate the probability mass around zero as much as possible. Dwork et al. [11] proposed the Laplace mechanism to preserve differential privacy for numerical values by calibrating the standard deviation of the noise according to the global sensitivity of the query function. Almost all the work done in the context of differential privacy for numerical values has used the Laplace mechanism to achieve differential privacy guarantees. However, there is little work to find the optimal data-independent noise distribution to achieve differential privacy [14,31]. For example, Soria-Comas et al. [31] proposed a general optimality criterion based on the concentration of the probability mass of the noise distribution around zero. They showed that any noise optimal under this criterion must be optimal under any other sensible criterion. They also built the optimal data-independent noise distribution. Geng et al. [14] derived the optimal ε-differentially private mechanism for single real-valued query functions under a very general utility maximization (or cost minimization) framework. They showed that the class of noise probability distributions in the optimal mechanism has staircase-shaped probability density functions that are symmetric (around the origin), monotonically decreasing, and geometrically decaying. Accordingly, in our work, optimal differential privacy can be achieved by applying the optimal noise distribution instead of the Laplace distribution.

In the last few years, some mechanisms have been proposed to enforce differential privacy in moving objects databases. For the first time, Chen et al. [3] studied the problem of differential privacy for moving objects databases. They proposed a data-dependent sanitization algorithm by constructing a noisy prefix tree over the underlying moving objects database. However, with the growth of the noisy prefix tree, the number of trajectories falling into the same branch decreases quickly, resulting in poor utility. To address this problem, in subsequent work, Chen et al. [2] employed a variable-length n-gram model that extracts the essential information in terms of a set of variable-length n-grams. The model makes use of an exploration tree based on the Markov assumption to decrease the magnitude of added noise. However, this work still suffers from the problem that by increasing the number of locations, the size of the exploration tree will grow exponentially, and thus, it is not scalable for spatial domains with a large number of locations. He et al. [17] presented DPT, a system to synthesize trajectories while ensuring differential privacy. DPT, which stands for differentially private trajectories, discretizes the spatial domain at multiple resolutions using a hierarchy of reference systems to capture movements at different speeds. However, DPT suffers from the problem that, for fine resolutions, the frequencies of subtrajectories will be small, and thus, the added noise will become relatively large. Wang et al. [34] proposed a private trajectories calibration and publication system (PTCP), which can be used to release trajectories in social media under differential privacy. PTCP adopts a noisy calibrated trajectories publication solution with privacy guarantees by building noise-enhanced prefix trees and extends the utility of released data through a differentially private post-processing sampling approach. However, all of these works use some tree structure to represent a moving objects database that causes the noise added to nodes with small real value results in a large relative error. Moreover, leveraging tree structures to represent moving objects databases usually incurs high time and space overheads. In this paper, we preserve the mobility patterns of original trajectories using a so-called normalized frequency matrix, which reduces time and space overheads.

Li et al. [23] proposed a differentially private trajectory data release mechanism with a bounded noise generation and a trajectory merging algorithm. The noise generation algorithm is designed such that the noise added to true trajectory counts is sampled in a legal range. Xu et al. [36] proposed DP-LTOD, a differential privacy latent trajectory community discovering scheme, which obfuscates original trajectory sequences into differentially private trajectory sequences. DP-LTOD first partitions an original trajectory sequence into different segments. Then, it selects the suitable locations and segments to constitute an obfuscated trajectory sequence. Specifically, it formulates a trajectory obfuscation problem to select an optimal trajectory sequence which has the smallest difference with the original trajectory sequence. Wang et al. [33] proposed DP-PSP, a differentially private statistics publication mechanism for real-time trajectory streams. DP-PSP discovers sensitive anchor points and divides the road network into a number of segments. Each spatial location in a trajectory stream is then calibrated to its nearest anchor point to handle the heterogeneity of trajectories. DP-PSP allows users to specify their own dynamic privacy budget distribution to optimize their own privacy budget. It also presents a private k-nearest neighbor selection and perturbation algorithm to reduce the amount of perturbation distortion induced by adding random noise.

Gursoy et al. [16] presented AdaTrace, a utility-aware trajectory synthesizer with differential privacy guarantee. AdaTrace performs feature extraction, learning, and noise injection using a database of real trajectories. It then generates synthetic trajectories while preserving differential privacy, enforcing resilience to inference attacks, and upholding statistical and spatial utilities. They also presented DP-Star [15], similar work to AdaTrace, which uses a normalization algorithm to summarize raw trajectories using their representative points, in its first step. However, these works do not properly consider some useful properties of original trajectories, such as number of points and mobility patterns, in synthetic trajectories and, thus, cannot preserve some spatial and temporal properties of original trajectories (as we will show in our experiments). Moreover, they do not consider time-dependent locations and, thus, are not able to answer time-dependent queries.

Deldar and Abadi [8] presented PDP-SAG, a differentially private mechanism that combines the sensitive attribute generalization (SAG) with personalized differential privacy (PDP) in a unified manner. By this combination, they aimed to provide different levels of differential privacy protection for moving objects that have non-spatiotemporal sensitive attributes as well. However, this work aims to provide personalized differential privacy for moving objects databases that have non-spatiotemporal sensitive attributes as well and does not release synthetic moving objects databases, as we do in this paper.

3 Preliminaries

In this section, we give some definitions and preliminaries that are used throughout the paper.

3.1 Differential privacy

Differential privacy (DP) is one of the strongest privacy guarantees available today that provides a mathematically provable guarantee of privacy protection against a wide range of privacy attacks [12]. It guarantees that the adversary will learn no information about an individual data record, even though he/she observes sequences of query outputs from two neighboring databases, one with and the other without that data record. In the following, we define the concepts related to differential privacy.

Definition 1 (*Neighboring databases*) Two distinct databases $\mathcal{D}_1$ and $\mathcal{D}_2$ from the universe of databases $\mathfrak{D}$ are said to be neighbors, denoted by $\mathcal{D}_1 \sim \mathcal{D}_2$, iff one can be obtained by adding or removing a single data record from the other.

Definition 2 (*ε-Differential privacy*) A randomized algorithm $\mathcal{A}$ is said to be ε-differentially private or ε-DP iff for any two input neighboring databases $\mathcal{D}_1$ and $\mathcal{D}_2$, and any subset O of all possible outputs of $\mathcal{A}$, we have

$$\Pr[\mathcal{A}(\mathcal{D}_1) \in O] \leq \exp(\varepsilon) \times \Pr[\mathcal{A}(\mathcal{D}_2) \in O], \tag{1}$$

where ε is a privacy parameter, known as the total privacy budget, that determines the strength of the privacy guarantee. A smaller ε will result in a stronger privacy guarantee, and vice versa.

A popular and widely used mechanism for answering statistical queries under differential privacy is the Laplace mechanism [11], which adds random noise drawn from the Laplace distribution to the output of statistical queries. The magnitude of the noise is scaled according to the (global) sensitivity of the query function, which is a measure of the maximum possible change to query outputs over any two neighboring databases.

Definition 3 (*Sensitivity*) Let $f : \mathfrak{D} \to \mathbb{R}^d$ be a query function that maps any database in the universe of databases $\mathfrak{D}$ to a vector of d real numbers. The sensitivity of f, denoted by σ_f, is defined as

$$\sigma_f = \max_{\mathcal{D}_1 \sim \mathcal{D}_2} \| f(\mathcal{D}_1) - f(\mathcal{D}_2) \|_1, \tag{2}$$

where $\| \cdot \|_1$ denotes the L^1-norm of a vector.

Definition 4 (*Laplace mechanism*) Let $f : \mathfrak{D} \to \mathbb{R}^d$ be a query function for the universe of databases $\mathfrak{D}$. A randomized algorithm $\mathcal{A}$ satisfies ε-DP iff for any input database $\mathcal{D} \in \mathfrak{D}$, we have

$$\mathcal{A}(\mathcal{D}) = f(\mathcal{D}) + \mathrm{Lap}(\sigma_f/\varepsilon), \tag{3}$$

where $\mathrm{Lap}(\lambda)$ is a Laplace random variable with probability density function $h_\lambda(z) = \frac{1}{2\lambda} \exp(-|z|/\lambda)$ and variance $2\lambda^2$.

The Laplace mechanism does not apply to all statistical queries, such as those that have categorical (or discrete) outputs. The exponential mechanism [25] is more general than the Laplace mechanism and applies to all types of queries. It uses an arbitrary scoring function that given an input database $\mathcal{D}$ and a discrete output r, it assigns a real-valued score to r to quantify the quality of r.

Definition 5 (*Exponential mechanism*) Let $q : \mathfrak{D} \times \mathcal{R} \to \mathbb{R}$ be an arbitrary scoring function for the universe of databases $\mathfrak{D}$ and a domain of discrete outputs $\mathcal{R}$. The randomized algorithm $\mathcal{A}$ that returns the discrete output $r \in \mathcal{R}$ for an input database $\mathcal{D} \in \mathfrak{D}$ with a probability proportional to $\exp(\varepsilon q(\mathcal{D}, r)/2\sigma_q)$ satisfies ε-DP, where σ_q is the sensitivity of q and defined as

$$\sigma_q = \max_{r \in \mathcal{R}, \mathcal{D}_1 \sim \mathcal{D}_2} \| q(\mathcal{D}_1, r) - q(\mathcal{D}_2, r) \|_1. \tag{4}$$

Any sequence of differential privacy computations is also differentially private. This property is known as compositionality and has two different types: sequential composition and parallel composition [26].

Theorem 1 *Let $\Lambda = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n\}$ be a set of randomized algorithms, where each $\mathcal{A}_i \in \Lambda$ satisfies ε_i-DP for an input database $\mathcal{D}$. Then, the sequential composition $\mathcal{A}_1 \circ \mathcal{A}_2 \circ \cdots \circ \mathcal{A}_n$ over $\mathcal{D}$ satisfies $(\sum_i \varepsilon_i)$-DP and the parallel composition $\mathcal{A}_1 \parallel \mathcal{A}_2 \parallel \cdots \parallel \mathcal{A}_n$ over disjoint subsets of $\mathcal{D}$ satisfies* $(\max_i \varepsilon_i)$*-DP* [26].

As mentioned above, differential privacy guarantees that the distribution of query results changes only slightly due to the modification of any one data record in the database. This allows protection against powerful adversaries who know the entire database except for one data record. On the other hand, differential privacy mechanisms implicitly assume that data records in a database are independent. To the best of our knowledge, all of the works that apply differential privacy to real databases such as moving objects databases also follow this assumption [15–17]. Similarly, we follow the same assumption in this paper. However, few works have introduced the issue of dependable data records in differential privacy [21,24]. As discussed in [21], if we do not make this assumption and consider a database where some individuals may have multiple data records; according to the Pareto principle, most of the individuals in this database will have few data records (or often one data record), whereas a small proportion of them may have more data records (see [21] for more details). Thus, we can separate a large number of low-frequency individuals from a small number of high-frequency ones and compute the sensitivity of the query function based on the group of low-frequency individuals [21]. This allows us to guarantee ε-DP for most individuals while having a little weaker differential privacy guarantee for others.

3.2 Moving objects database

Moving objects databases store and manage discrete or continuous changes of moving objects over an underlying spatial domain.

Given a spatial domain in which the movement of moving objects is constrained within it, a moving objects database $\mathcal{D}$ over this spatial domain is a multiset of trajectories. Each trajectory $T \in \mathcal{D}$ is a sequence of points or latitude/longitude locations $\langle X_1, X_2, \ldots, X_{|T|}\rangle$, where $|T|$ is the length (or number of points) of T. The point X_1 is called the head of T, and the subtrajectory $\langle X_2, X_3, \ldots, X_{|T|}\rangle$ is called the tail of T. More specifically, the head of T is defined to be its leading point, and the tail of T is defined to be the subtrajectory obtained by removing its leading point.

Definition 6 (*Subtrajectory*) A trajectory $T_r = \langle X_1^r, X_2^r, \ldots, X_n^r\rangle$ is said to be a subtrajectory of a trajectory $T_s = \langle X_1^s, X_2^s, \ldots, X_m^s\rangle$, iff there exists n consecutive integers $1 \leq i < i+1 < \cdots < i+n-1 \leq m$ such that $X_1^r = X_i^s, X_2^r = X_{i+1}^s, \ldots, X_n^r = X_{i+n-1}^s$.

4 Differentially private moving objects database release

Many companies like Google, Uber, and others collect a huge volume of data about the movements of moving objects every day through their mobile apps, resulting in large moving objects databases. Analyzing such databases is of great value for data analysts and has many applications in different tasks such as city planning, traffic analysis, taxi service prediction, and passenger demand analysis. However, due to the concerns of disclosure of any information about moving objects such as trip histories, home and work locations, frequent meeting points, or visits to sensitive locations like hospitals, health clinics, and airports, these companies often cannot safely provide their collected moving objects databases to data analysts.

In this section, we introduce DP-MODR, a differentially private mechanism for synthetic moving objects database release that preserves spatial and temporal utilities efficiently and effectively. DP-MODR consists of five main steps. In the first step, we discretize the continuous spatial domain into a finite set of domain cells and create a noisy histogram of starting domain cells of original trajectories to keep the distribution of trajectory heads. In the second step, we compute the noisy median length of original trajectories that start in each domain cell to preserve the distribution of trajectory lengths around their median. In the third step, we construct a noisy transition cost matrix to preserve the mobility patterns of original trajectories. In the fourth step, we construct some noisy cost-sensitive path trees using the noisy transition cost matrix to keep existing most probable domain cell paths with different lengths and different starting domain cells. Finally, in the fifth step, we release synthetic trajectories by constructing a differentially private moving objects database using the information obtained in the previous steps. It should be mentioned that the first three steps work on original trajectories; therefore, to satisfy differential privacy, we divide ε into three parts, namely ε_1, ε_2, and ε_3, and give each part to one of the steps, respectively. For the rest of this section, we will assume that we are given a moving objects database $\mathcal{D}$, and our goal is to release a differentially private version of it, denoted by $\hat{\mathcal{D}}$. Table 1 summarizes the notations used throughout the paper.

4.1 Creating a noisy starting domain cells histogram

In this step, we first discretize the continuous spatial domain into a finite set of regions or domain cells $\mathcal{C}$. The points of each trajectory in $\mathcal{D}$ are then mapped to the domain cells covering them. After that, we create a histogram of starting domain cells (SDCs) of all trajectories in $\mathcal{D}$. Note that by the starting domain cell or SDC of a trajectory, we mean the domain cell that covers the head of that trajectory. Besides, we add Laplace noise with scale parameter $1/\varepsilon_1$ to each bin of the obtained histogram independently. This results in a noisy histogram, also known as noisy starting domain cells histogram or NSDC histogram. By adding/removing one trajectory to/from $\mathcal{D}$, the value of exactly one bin will be

Table 1 Notations used throughout the paper

Symbol	Description
$\mathcal{D}$	A moving objects database
$\hat{\mathcal{D}}$	A synthetic moving objects database
T	A trajectory
$\hat{T}$	A synthetic trajectory
C_i	A domain cell
$\mathcal{C}$	A set of domain cells
ε	The total privacy budget
H	An NSDC histogram
θ_{C_i}	The median length of trajectories starting in a domain cell C_i
$\hat{\theta}_{C_i}$	The noisy median length of trajectories starting in a domain cell C_i
$\bar{c}_{\mathcal{D}}$	A normalized frequency function over a moving objects database $\mathcal{D}$
$\mathbf{F}$	A normalized frequency matrix
$\mathbf{C}$	A noisy transition cost matrix
$h_{\max}$	The maximum height of a noisy cost-sensitive path tree
Φ_{C_i}	A noisy cost-sensitive path tree for a domain cell C_i
ϑ	A cost function
$l_{\max}$	The maximum possible length of a trajectory

changed by 1. Therefore, from Definition 4, we conclude that the NSDC histogram is ε_1-differentially private or ε_1-DP.

In fact, this step aims to keep the distribution of the starting points of original trajectories as much as possible, which is one of the useful properties of moving objects databases. By preserving this property, the synthetic moving objects database can be effectively used to answer popular queries such as "number of trajectories that have the same starting point." Such queries can have many applications in urban management and traffic analysis issues.

Example 1 Consider the moving objects database of Table 2 that has been constructed over a discretized spatial domain with domain cells C_1, C_2, C_3, and C_4. Figure 1a shows the SDC histogram of this moving objects database. Also, Fig. 1b shows an NSDC histogram that is a noisy version of the SDC histogram of Fig. 1a.

4.2 Estimating noisy trajectory lengths

In this step, for each domain cell $C_i \in \mathcal{C}$, we compute the median length of trajectories starting in C_i and then run a differentially private mechanism (with ε_2 as the privacy parameter) on the obtained median lengths. This allows us to preserve the length of original trajectories when generating synthetic trajectories for public release, which is one of the important statistical properties of moving objects databases. By preserving this property, we can generate synthetic trajectories of lengths as closely as possible to the original ones. Similar to the previous step, we can obtain a noisy median length by perturbing it with Laplace noise, but we know that the median function has a large sensitivity for trajectory lengths, and therefore, adding Laplace noise destroys data utility. Hence, similar to prior work [20,22], we choose the exponential mechanism instead.

Table 2 A moving objects database

ID	Trajectory
1	$\langle C_1, C_4, C_4\rangle$
2	$\langle C_2, C_1, C_2, C_2, C_4\rangle$
3	$\langle C_1, C_2, C_4, C_4, C_3, C_2\rangle$
4	$\langle C_2, C_1, C_3, C_2, C_3, C_2\rangle$
5	$\langle C_4, C_2, C_3, C_2\rangle$
6	$\langle C_1, C_3, C_2, C_4, C_2\rangle$
7	$\langle C_2, C_2, C_4, C_3\rangle$

To obtain the noisy median lengths using the exponential mechanism, we proceed as follows. For each domain cell $C_i \in \mathcal{C}$, we first create a multiset L of trajectory lengths whose corresponding trajectories start in C_i. Then, we sort L in non-decreasing order and choose a trajectory length $l \in L$ as the noisy median length $\hat{\theta}_{C_i}$ with a probability proportional to $\exp(\varepsilon_2 s(l, \theta_{C_i})/2)$, where θ_{C_i} is the median of trajectory lengths in L and $s(l, \theta_{C_i})$ is the score of l with respect to θ_{C_i}:

$$s(l, \theta_{C_i}) = -|r(l) - r(\theta_{C_i})|, \tag{5}$$

where $r(\cdot)$ returns the rank of a trajectory length in L. The intuition is that if a trajectory length is close to θ_{C_i}, then its rank will be similar to the rank of θ_{C_i}. Thus, the score of each candidate trajectory length should be negatively proportional to the absolute difference between its rank and the rank of θ_{C_i}.

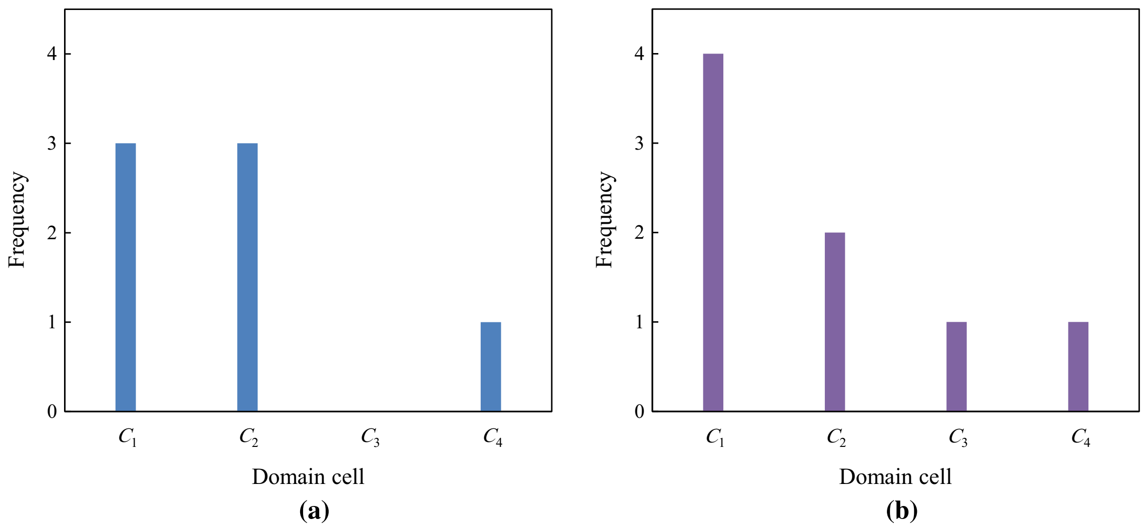

Fig. 1 SDC and NSDC histograms of the moving objects database of Table 2: **a** SDC histogram. **b** NSDC histogram

This causes trajectory lengths that are closer to θ_{C_i} to have a higher probability of being chosen as the noisy median length.

By adding/removing one trajectory to/from $\mathcal{D}$, the rank of only one median length changes by at most 1. This shows that the sensitivity of the scoring function s is 1. Therefore, from Definition 5, we conclude that the obtained noisy median lengths are ε_2-differentially private or ε_2-DP.

Example 2 Consider the moving objects database of Table 2 and suppose that ε_2 is 0.40. To obtain the noisy median length of trajectories starting in C_1, namely $\hat{\theta}_{C_1}$, we first create the multiset L and sort it in non-decreasing order; therefore, we obtain $L = \{3, 5, 6\}$. Since $\theta_{C_1} = 5$, we have $s(3, 5) = -1$, $s(5, 5) = 0$, and $s(6, 5) = -1$. Thus, we set $\hat{\theta}_{C_1}$ to one of the values 3, 5, or 6 with a probability of 0.31, 0.38, or 0.31, respectively.

4.3 Constructing a noisy transition cost matrix

In this step, we construct a noisy transition cost matrix $\mathbf{C} = (c_{ij})_{m\times m}$ over $\mathcal{D}$ to preserve the mobility patterns of trajectories in $\mathcal{D}$, where m is the number of domain cells in $\mathcal{C}$. It is worth mentioning that mobility patterns are the most important properties of moving objects databases, and thus, preserving them in synthetic moving objects databases is so important. By preserving these properties, the synthetic moving objects database can be effectively used for different data analysis tasks, including count query answering and frequent pattern mining. Each element $c_{ij} \in \mathbf{C}$ stores the noisy cost of transition from a domain cell $C_i \in \mathcal{C}$ to a domain cell $C_j \in \mathcal{C}$:

$$c_{ij} = -\log p_{ij}, \tag{6}$$

where p_{ij} is the differentially private noisy transition probability (or just noisy transition probability in short) from C_i to C_j.

To obtain noisy transition probabilities, we first construct a normalized frequency matrix $\mathbf{F} = (f_{ij})_{m\times m}$ over $\mathcal{D}$, whose rows and columns are uniquely labeled with the domain cells of $\mathcal{C}$. Each element $f_{ij} \in \mathbf{F}$ stores the normalized frequency of a subtrajectory $\langle C_i, C_j\rangle$ in $\mathcal{D}$, where C_i and C_j belong to $\mathcal{C}$.

Definition 7 (*Normalized frequency*) Let $\mathcal{C}$ be the set of all domain cells within a given spatial domain and $\mathcal{D}$ be an arbitrary moving objects database. The normalized frequency of a subtrajectory $\langle C_i, C_j\rangle$ in $\mathcal{D}$, denoted by $\bar{c}_{\mathcal{D}}(\langle C_i, C_j\rangle)$, is defined as

$$\bar{c}_{\mathcal{D}}(\langle C_i, C_j\rangle) = \sum_{T\in\mathcal{D}} \frac{c_T(\langle C_i, C_j\rangle)}{|T| - 1}, \tag{7}$$

where T is an arbitrary trajectory of $\mathcal{D}$ and $c_T(\langle C_i, C_j\rangle)$ is the frequency of the subtrajectory $\langle C_i, C_j\rangle$ in T.

Example 3 Consider the moving objects database of Table 2. The normalized frequency matrix $\mathbf{F}$ for this moving objects database is constructed as

$$\mathbf{F} = \begin{bmatrix} 0 & 0.45 & 0.45 & 0.50 \\ 0.45 & 0.58 & 0.53 & 1.03 \\ 0 & 1.18 & 0 & 0 \\ 0 & 0.58 & 0.53 & 0.70 \end{bmatrix}.$$

Subsequently, we perturb the elements of $\mathbf{F}$ by adding Laplace noise with scale parameter $1/\varepsilon_3$ to them. The noisy transition probabilities are then computed by normalizing the rows of $\mathbf{F}$ to sum up to 1. More formally,

$$p_{ij} = \frac{\hat{f}_{ij}}{\sum_j \hat{f}_{ij}}, \tag{8}$$

where $\hat{f}_{ij}$ is the noisy version of an element $f_{ij} \in \mathbf{F}$.

It is worth mentioning that the sensitivity of the normalized frequency function $\bar{c}_{\mathcal{D}}$ in (7) is 1. This is because the frequency of each pair of domain cells in a desired trajectory T is bounded by $|T| - 1$. Therefore, by adding/removing T to/from $\mathcal{D}$, $\bar{c}_{\mathcal{D}}$ changes by at most 1. Therefore, from Definition 4, we conclude that the obtained noisy transition probabilities are ε_3-differentially private or ε_3-DP.

4.4 Constructing noisy cost-sensitive path trees

In this step, we define a so-called noisy cost-sensitive path tree of the maximum height $h_{\max}$ for each domain cell of the underlying spatial domain to keep all existing most probable paths with different lengths (up to $h_{\max} + 1$) starting in that domain cell. The noisy cost-sensitive path trees are constructed to keep all existing most probable paths efficiently in terms of time and space complexities. We use these cost-sensitive path trees to construct synthetic trajectories in such a way that their mobility patterns are as similar as possible to the mobility patterns of original trajectories.

Definition 8 (*Noisy cost-sensitive path tree*) Let $\mathcal{C}$ be a set of m domain cells defined over a given spatial domain. A noisy cost-sensitive path tree for an arbitrary domain cell $C_i \in \mathcal{C}$ is a triple $\Phi_{C_i} = (V, E, \vartheta)$, where V is the set of nodes, E is the set of edges, and $\vartheta : V \rightarrow \mathbb{R}_{\geq 0}$ is a cost function. Each level of Φ_{C_i} contains at most m nodes, each of which is uniquely labeled with one domain cell in $\mathcal{C}$. The root of Φ_{C_i} is at level 0 and labeled with C_i. The cost function for each node at level l gives the cost of the most probable path with length $l + 1$ (if exists) from C_i to the domain cell labeling that node.

We use an efficient bottom-up dynamic programming algorithm (Algorithm 1) to construct a noisy cost-sensitive path tree Φ_{C_i} for any given domain cell $C_i \in \mathcal{C}$. In this algorithm, we first create the root r of Φ_{C_i} at level 0, label r by C_i, and set its cost to 0. Then, we create other nodes of Φ_{C_i} in a breadth-first order until the level $h_{\max}$ is reached or Φ_{C_i} cannot be expanded further. For each level, we consider m potential nodes and label each one with a unique domain cell in $\mathcal{C}$. We then compute the cost $\vartheta(v)$ for each potential node v at this level as

$$\vartheta(v) = \min_{u \in L_V(v)} \vartheta(u) + c(u, v), \tag{9}$$

Algorithm 1 Noisy cost-sensitive path tree construction

```
Input:
    $C_i$: A domain cell
Output:
    $\Phi_{C_i}$: A noisy cost-sensitive path tree

Create the root $r$ of $\Phi_{C_i}$ at level 0
Label $r$ by $C_i$ and set its cost to 0
for each level of $\Phi_{C_i}$ from 1 up to $h_{max}$ do
    for each domain cell $C_j \in \mathcal{C}$ do
        Create a node $v$ and label it by $C_j$
        Compute the cost $\vartheta(v)$ using (9)
        if $\vartheta(v) \neq \infty$ then
            Compute the parent $\eta(v)$ using (10)
            Add $v$ to $V(\Phi_{C_i})$ and $(\eta(v), v)$ to $E(\Phi_{C_i})$
        end if
    end for
end for
for each level $l$ of $\Phi_{C_i}$ starting from the last level do
    for each leaf node $v$ at level $l$ do
        if $\vartheta(v)$ is not minimum at level $l$ then
            Eliminate $v$ from $V(\Phi_{C_i})$
            Eliminate $(\eta(v), v)$ from $E(\Phi_{C_i})$
        end if
    end for
end for
```

where $L_V(v)$ is the set of all nodes in the immediately preceding level (i.e., the level preceding the level of v) and $c(u, v)$ is the noisy cost of transition from the labeling domain cell of u to the labeling domain cell of v (refer to Sect. 4.3 for more information). We also find the parent of v as

$$\eta(v) = \arg\min_{u \in L_V(v)} \vartheta(u) + c(u, v). \tag{10}$$

If $\vartheta(v) \neq \infty$, then this means that there is at least one path from C_i to the labeling domain cell of v. In this case, we add the node v to $V(\Phi_{C_i})$ and the edge $(\eta(v), v)$ to $E(\Phi_{C_i})$.

After the construction of Φ_{C_i} is completed, we prune it by eliminating those nodes that do not exist in any most probable path. To do this, we traverse Φ_{C_i}, level by level starting from the last level, and eliminate the leaf nodes at each level whose associated cost is not minimum at that level. In this way, we keep the minimum number of nodes required.

Example 4 Let us assume that the noisy transition cost matrix for the moving objects database of Table 2 is constructed as

$$\mathbf{C} = \begin{bmatrix} 1.30 & 0.52 & 0.46 & 0.52 \\ 1.00 & 0.70 & 0.70 & 0.30 \\ 1.22 & 0.06 & 1.40 & 1.30 \\ 1.30 & 0.70 & 0.60 & 0.35 \end{bmatrix}.$$

Figure 2 shows the noisy cost-sensitive path trees Φ_{C_1}, Φ_{C_2}, Φ_{C_3}, and Φ_{C_4} by fixing the maximum height $h_{\max}$ to 2. For each node, its labeling domain cell is placed inside and

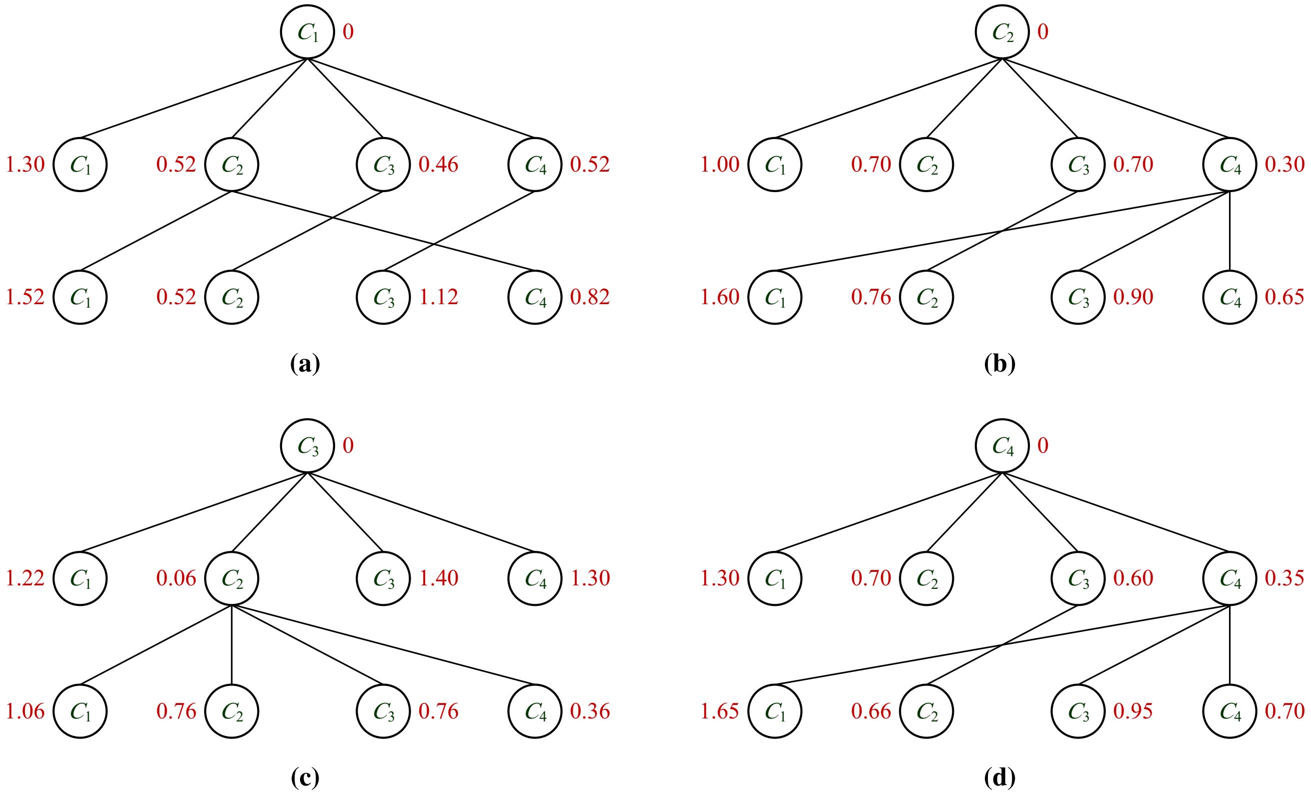

Fig. 2 Noisy cost-sensitive path trees by fixing the maximum height $h_{\max}$ to 2: **a** Φ_{C_1}. **b** Φ_{C_2}. **c** Φ_{C_3}. **d** Φ_{C_4}

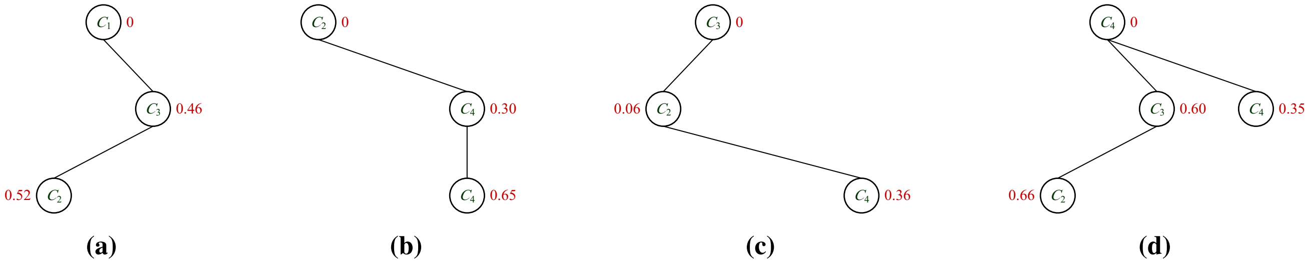

Fig. 3 Pruned noisy cost-sensitive path trees: **a** Φ_{C_1}. **b** Φ_{C_2}. **c** Φ_{C_3}. **d** Φ_{C_4}

its cost is placed outside that node. Figure 3 shows the pruned version of each of these noisy cost-sensitive path trees, where nodes that do not exist in any most probable path have been eliminated.

We can use Φ_{C_i} to find the most probable path of any given length starting in C_i. Specifically, to find the most probable path with length $l + 1$ starting in C_i, we start from the root of Φ_{C_i}, r, and follow the path that leads to a node with the minimum cost at level l of Φ_{C_i}. We refer to such a path as a most probable root-originated path.

Example 5 Consider the noisy cost-sensitive path tree Φ_{C_4} of Fig. 3. The most probable root-originated path with length 3 in Φ_{C_4} is $\langle C_4, C_3, C_2 \rangle$.

4.5 Constructing a differentially private moving objects database

In this step, we construct a differentially private moving objects database consisting of synthetic trajectories. We do this as described below. Let $l_{\max}$ be the maximum possible length of trajectories. First, we initialize the synthetic moving objects database $\hat{\mathcal{D}}$ to be an empty set. Then, for each domain cell $C_i \in \mathcal{C}$, we repeat the following procedure as many times as the value of the corresponding bin of C_i in the NSDC histogram. At each iteration, we draw a sample l from an exponential distribution with parameter $\ln(2)/\hat{\theta}_{C_i}$ and limit it to $l_{\max}$, where $\hat{\theta}_{C_i}$ is the noisy median length of trajectories starting in C_i. We then generate a synthetic trajectory $\hat{T}$ with length l starting in C_i. To do so, we first

Algorithm 2 Differentially private moving objects database construction

```
Input:
    $\mathcal{C}$: A set of domain cells
    $H$: An NSDC histogram
Output:
    $\hat{\mathcal{D}}$: A synthetic moving objects database

Initialize $\hat{\mathcal{D}}$ to be an empty set
for each domain cell $C_i \in \mathcal{C}$ do
    Let $H(C_i)$ be the value of the corresponding bin of $C_i$ in $H$
    for $k = 1$ to $H(C_i)$ do
        Let $\hat{\theta}_{C_i}$ be the noisy median length of trajectories starting in $C_i$
        Draw a sample $l$ from an exponential distribution with parameter $\ln(2)/\hat{\theta}_{C_i}$
        Create a synthetic trajectory $\hat{T}$ and initialize it to $\langle C_i \rangle$
        while $|\hat{T}| < l$ do
            Let $E$ be the ending domain cell of $\hat{T}$ and $h$ be the height of the cost-sensitive path tree $\Phi_E$
            Find the most probable root-originated path $P$ with length $\min(l - |\hat{T}| + 1, h + 1)$ in $\Phi_E$
            Append the tail of $P$ to $\hat{T}$
        end while
        Replace each domain cell of $\hat{T}$ by a latitude/longitude location
        Add $\hat{T}$ to $\hat{\mathcal{D}}$
    end for
end for
```

initialize $\hat{T}$ to $\langle C_i \rangle$. Then, we repeatedly start from the ending domain cell E of $\hat{T}$, find the most probable root-originated path P with length $h+1$ starting in E (using the noisy cost-sensitive path tree Φ_E), and append the tail of P to $\hat{T}$, where h is the height of Φ_E. We repeat this until the difference between l and $|\hat{T}|$ becomes lower than or equal to h. At this time, we find the most probable root-originated path with length $l - |\hat{T}| + 1$ in Φ_E and append its tail to $\hat{T}$. Finally, we replace each domain cell of $\hat{T}$ by a latitude/longitude location that is covered by that domain cell (e.g., the centroid latitude/longitude location of the domain cell) and then add $\hat{T}$ to $\hat{\mathcal{D}}$. Algorithm 2 shows the pseudocode of the procedure described above.

Example 6 Consider the pruned noisy cost-sensitive path trees of Fig. 3. To generate a synthetic trajectory $\hat{T}$ with length $l = 6$ starting in C_1, we first initialize $\hat{T}$ to $\langle C_1 \rangle$. Then, we find the most probable root-originated path with length 3 in Φ_{C_1} and append its tail to $\hat{T}$, resulting in $\hat{T} = \langle C_1, C_3, C_2 \rangle$. Subsequently, we find the most probable root-originated path with length 3 in Φ_{C_2} and append its tail to $\hat{T}$, resulting in $\hat{T} = \langle C_1, C_3, C_2, C_4, C_4 \rangle$. Finally, we find the most probable root-originated path with length 2 in Φ_{C_4} and append its tail to $\hat{T}$. Therefore, we obtain $\hat{T} = \langle C_1, C_3, C_2, C_4, C_4, C_4 \rangle$.

4.6 Privacy analysis

In the following, we analyze the privacy guarantee of DP-MODR and prove that it satisfies ε-differential privacy or ε-DP.

Theorem 2 *DP-MODR satisfies ε-DP.*

Proof As described before, DP-MODR consists of five main steps. We have already shown that the first, second, and third steps satisfy ε_1-DP, ε_2-DP, and ε_3-DP, respectively. On the other hand, the fourth and fifth steps work on the differentially private (noisy) outputs of the first three steps, and thus, each of them satisfies 0-DP. Thus, from Theorem 1, we conclude that DP-MODR satisfies $(\varepsilon_1 + \varepsilon_2 + \varepsilon_3)$-DP or ε-DP. □

4.7 Performance analysis

In the following, we analyze the efficiency of DP-MODR in terms of time and space complexities.

Let n be the number of trajectories in $\mathcal{D}$. As described before, DP-MODR consists of five main steps. In the first step, DP-MODR traverses $\mathcal{D}$ once to create an NSDC histogram, which takes $O(n)$ time. In the second step, DP-MODR computes the noisy median length of original trajectories starting in each domain cell, which takes as much time as the number of these trajectories. Thus, this step takes $O(n)$ time for all domain cells. In the third step, DP-MODR constructs a noisy transition cost matrix. This task takes $O(l_{\max} \times n + m^2)$ time, where $l_{\max}$ is the maximum possible length of a trajectory and m is the number of domain cells. In the fourth step, DP-MODR constructs noisy cost-sensitive path trees. Since the number of nodes in each noisy cost-sensitive path tree is at most $h_{\max} \times m$ and the cost computation for each node takes $O(m)$ time, the construction of all m pruned noisy cost-sensitive path trees takes $O(h_{\max} \times m^3)$ time. Finally, in the fifth step, DP-MODR releases synthetic trajectories. Since the length of each synthetic trajectory is assumed to be at most $l_{\max}$ and the number of all synthetic trajectories is approximately as same as the number of original trajectories, this step takes $O(l_{\max} \times n)$ time. As a result, we conclude that the overall time complexity of DP-MODR is $O(l_{\max} \times n + h_{\max} \times m^3)$. Since we usually set $h_{\max}$ to a small value, it becomes $O(l_{\max} \times n + m^3)$.

Moreover, from the above discussion, it is clear that the space complexity of DP-MODR is equal to the total space required to store the NSDC histogram, the noisy median lengths, the noisy cost matrix, and the pruned noisy cost-sensitive path trees, which are $O(m)$, $O(m)$, $O(m^2)$, and $O(h_{\max} \times m^2)$, respectively. Therefore, we conclude that the overall space complexity of DP-MODR is $O(m^2)$.

5 Differentially private time-dependent moving objects database release

In this section, we explain how to extend DP-MODR to support moving objects databases whose locations are time-dependent. We refer to such databases as time-dependent moving objects databases. The new differentially private mechanism, also known as DP-MODRT, is especially suitable for answering time-dependent queries. An example of such queries in the real world is "number of trajectories that have the same starting and ending points at around the same time."

Let $\mathcal{D}$ be a time-dependent moving objects database. Each point in a trajectory $T \in \mathcal{D}$ is denoted by a pair (X_i, t_i), where X_i is the latitude/longitude location of the owner of T at time stamp t_i. Therefore, two trajectories with the same sequence of locations but different time stamps are considered different. In the first step of DP-MODRT, in addition to discretizing the continuous spatial domain into a finite set of regions or domain cells, we also discretize the time space into non-overlapping time intervals. Consequently, we consider $\mathcal{C}$ to be a set of time-dependent domain cells. In other words, all combinations of domain cells and time intervals are included in $\mathcal{C}$. Then, we create a noisy starting domain cells histogram of all trajectories in $\mathcal{D}$ according to the time-dependent domain cells in $\mathcal{C}$.

Example 7 Consider the time-dependent moving objects database of Table 3 that has been constructed over a discretized spatial domain with domain cells C_1, C_2, C_3, and C_4, and time intervals 1, 2, and 3. Figure 4 shows the SDC histogram of this moving objects database.

Table 3 A time-dependent moving objects database

ID	Trajectory
1	$\langle (C_1, 1), (C_4, 1), (C_4, 2) \rangle$
2	$\langle (C_2, 1), (C_1, 2), (C_2, 2), (C_2, 3), (C_4, 3) \rangle$
3	$\langle (C_1, 1), (C_2, 1), (C_4, 2), (C_4, 2), (C_3, 3), (C_2, 3) \rangle$
4	$\langle (C_2, 1), (C_1, 1), (C_3, 1), (C_2, 2), (C_3, 2), (C_2, 3) \rangle$
5	$\langle (C_4, 2), (C_2, 3), (C_3, 3), (C_2, 3) \rangle$
6	$\langle (C_1, 2), (C_3, 2), (C_2, 2), (C_4, 3), (C_2, 3) \rangle$
7	$\langle (C_2, 2), (C_2, 2), (C_4, 2), (C_3, 3) \rangle$

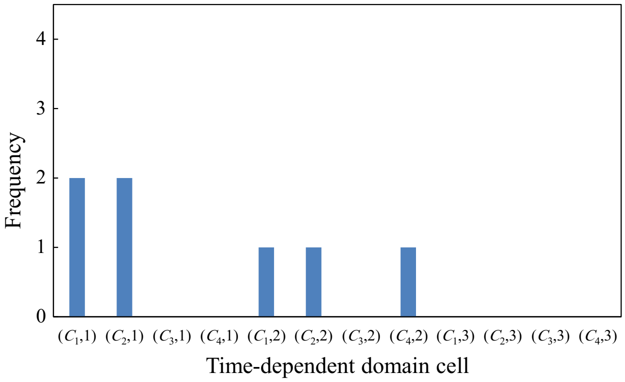

Fig. 4 SDC histogram of the time-dependent moving objects database of Table 3

Subsequently, for each time-dependent domain cell in $\mathcal{C}$, we compute the median length of trajectories starting in that domain cell in a differentially private way and then construct a normalized frequency matrix $\mathbf{F} = (f_{ij})_{m \times m}$, where m is the number of time-dependent domain cells in $\mathcal{C}$. It is clear that $\mathbf{F}$ is an almost upper triangular matrix because the time stamps in each trajectory are non-decreasing. On the other hand, for many pairs of time-dependent domain cells, it may not be possible to have a transition (due to their corresponding time intervals). Given this, and for the sake of simplicity, we assume that the time interval between any two consecutive points in $\mathcal{D}$ is less than or equal to the length of time intervals in the discretized time space. Thus, if the current point of a trajectory is at a discretized time interval t, its next point can only be at a discretized time interval t or $t+1$. As a result, the normalized frequency matrix $\mathbf{F}$ is sparse, and we can process $\mathbf{F}$ and also the noisy transition cost matrix $\mathbf{C}$ by taking full advantage of sparse matrix processing techniques. Finally, after constructing each synthetic trajectory, we replace each of its time-dependent domain cells by a latitude/longitude location that is covered by that time-dependent domain cell (e.g., the centroid latitude/longitude location) and a time stamp that is covered by the time interval associated with that time-dependent domain cell. Therefore, the transformed (differentially private) moving objects database can preserve the time information of trajectories (including the starting and ending time of points) as well as the location information, in a differentially private way.

6 Experiments

In this section, we empirically study the effectiveness of DP-MODR and DP-MODRT using real moving objects datasets.

6.1 Experimental setup

We perform our experiments using the following three real moving objects datasets:

- Geolife dataset. This dataset collects the GPS trajectories of some users during a period of over five years (from April 2007 to August 2012), recording a broad range of outdoor movements, including not only life routines like go home and go to work but also some entertainment and sports activities such as shopping, sightseeing, hiking, and cycling [38]. The majority of the trajectories are from Beijing, China, with few outliers from other cities in China, Europe, etc. We preprocess the dataset to remove some outliers; in particular, we choose the trajectories from Beijing whose points (latitude–longitude coordinates) are between [39.4, 40.8] latitude and [115.8, 117.4] longitude. In this way, we obtain approximately 17,000 trajectories.
- Taxi dataset.[1] This dataset contains approximately 1.7 million trajectories of taxis in Beijing, China, during a period of 9 days in May 2009. The trajectory data cover a region of Beijing restricted between [39.8, 40.1] latitude and [116.1, 116.6] longitude.
- Porto dataset. This dataset contains approximately 290,000 GPS trajectories of some taxis operating in Porto, Portugal, through a taxi dispatch central for one year (from July 2013 to July 2014). It is a subset of a larger dataset that was made available as part of the Taxi Service Trajectory Prediction Challenge at ECML-PKDD 2015 [27].

In our experiments, when constructing a synthetic moving objects database, we discretize the continuous spatial domain by partitioning it into a 32 × 32 grid, discretize the time space into 24 non-overlapping time intervals, and set the maximum height of noisy cost-sensitive path trees to $h_{\max} = 3$ (unless explicitly stated). Also, by default, we equally divide the total privacy budget ε among ε_1, ε_2, and ε_3.

Since the Laplace and exponential mechanisms are probabilistic, we repeat each experiment five times and report the average results. When computing the evaluation measures, we consider two different scenarios: the fine-grained scenario in which the continuous spatial domain is uniformly discretized into a finite set of 1024 domain cells and the coarse-grained scenario in which the continuous spatial domain is uniformly discretized into a finite set of 36 domain cells.

6.2 Evaluation measures

We use different measures to evaluate the spatial and temporal utilities of the synthetic moving objects databases constructed by DP-MODR.

6.2.1 Count query error

Let Q be a count query of the form “Retrieve the frequency of a given subtrajectory.” We define the relative error of the noisy answer to Q as

$$\mathcal{E}(Q) = \frac{\left|c_{\mathcal{D}}(Q) - c_{\hat{\mathcal{D}}}(Q)\right|}{\max\left\{c_{\mathcal{D}}(Q), \delta\right\}} \times 100, \tag{11}$$

where $c_{\mathcal{D}}(Q)$ and $c_{\hat{\mathcal{D}}}(Q)$ denote the true and noisy answers to Q when issued on $\mathcal{D}$ and $\hat{\mathcal{D}}$, respectively, and δ is a sanity bound used to mitigate the influences of count queries with extremely small true answers. Similar to most previous work [2,30], we set δ to be 0.1% of the number of trajectories in $\mathcal{D}$.

6.2.2 Locations rank correlation

The Kendall rank correlation coefficient, also referred to as Kendall's tau coefficient, is a statistic used to measure the rank correlation between two measured quantities. Intuitively, the Kendall rank correlation coefficient between two variables will be high when observations have a similar rank between two variables and will be low when observations have a dissimilar rank between two variables. We use the Kendall rank correlation coefficient to compute the locations rank correlation that measures the similarities and discrepancies between locations frequency ranking in $\mathcal{D}$ and $\hat{\mathcal{D}}$. Let $\mathcal{C}$ be the set of all domain cells (discrete locations) within the underlying discretized spatial domain. A pair of domain cells $C_i, C_j \in \mathcal{C}$ are said to be concordant if one of the following conditions holds:

$$\begin{aligned}&(1)\ c_{\mathcal{D}}(C_i) > c_{\mathcal{D}}(C_j) \ \text{ and } \ c_{\hat{\mathcal{D}}}(C_i) > c_{\hat{\mathcal{D}}}(C_j),\\&(2)\ c_{\mathcal{D}}(C_i) < c_{\mathcal{D}}(C_j) \ \text{ and } \ c_{\hat{\mathcal{D}}}(C_i) < c_{\hat{\mathcal{D}}}(C_j),\end{aligned} \tag{12}$$

where $c_{\mathcal{D}}(\cdot)$ and $c_{\hat{\mathcal{D}}}(\cdot)$ are the frequency of a given domain cell in $\mathcal{D}$ and $\hat{\mathcal{D}}$, respectively. They are said to be discordant if one of the following conditions holds:

$$\begin{aligned}&(1)\ c_{\mathcal{D}}(C_i) > c_{\mathcal{D}}(C_j) \ \text{ and } \ c_{\hat{\mathcal{D}}}(C_i) < c_{\hat{\mathcal{D}}}(C_j),\\&(2)\ c_{\mathcal{D}}(C_i) < c_{\mathcal{D}}(C_j) \ \text{ and } \ c_{\hat{\mathcal{D}}}(C_i) > c_{\hat{\mathcal{D}}}(C_j).\end{aligned} \tag{13}$$

If $c_{\mathcal{D}}(C_i) = c_{\mathcal{D}}(C_j)$ or $c_{\hat{\mathcal{D}}}(C_i) = c_{\hat{\mathcal{D}}}(C_j)$, they are neither concordant nor discordant. Briefly, two domain cells $C_i, C_j \in \mathcal{C}$ are concordant if their ranks in sorted order

[1] http://sensor.ee.tsinghua.edu.cn.

agree in $\mathcal{D}$ and $\hat{\mathcal{D}}$, and discordant otherwise. Accordingly, we define the locations rank correlation as

$$\tau_s = \frac{2}{m(m-1)} \left(\eta_s(\mathcal{D}, \hat{\mathcal{D}}) - \tilde{\eta}_s(\mathcal{D}, \hat{\mathcal{D}}) \right), \tag{14}$$

where m is the number of domain cells in $\mathcal{C}$. Also, $\eta_s(\mathcal{D}, \hat{\mathcal{D}})$ and $\tilde{\eta}_s(\mathcal{D}, \hat{\mathcal{D}})$ denote the number of concordant and discordant domain cell pairs, respectively.

6.2.3 Frequent patterns rank correlation

Let $\mathcal{F}$ be the set of top-k frequent patterns in $\mathcal{D}$. We use the Kendall rank correlation coefficient to compute the frequent patterns rank correlation that measures the similarities and discrepancies between the ranking of frequent patterns of $\mathcal{F}$ in $\mathcal{D}$ and $\hat{\mathcal{D}}$. We define the frequent patterns rank correlation as

$$\tau_t = \frac{2}{k(k-1)} \left(\eta_t(\mathcal{D}, \hat{\mathcal{D}}) - \tilde{\eta}_t(\mathcal{D}, \hat{\mathcal{D}}) \right), \tag{15}$$

where $\eta_t(\mathcal{D}, \hat{\mathcal{D}})$ and $\tilde{\eta}_t(\mathcal{D}, \hat{\mathcal{D}})$ denote the number of concordant and discordant frequent pattern pairs, respectively.

6.2.4 Trip error

One of the main properties of a trajectory is its starting and ending points that are mapped to regions or domain cells of the underlying discretized spatial domain. These points show a particular trip (e.g., a home-to-work commute or a taxi trip) and are important for many data analysis tasks such as city planning, taxi service prediction, and passenger demand analysis. We use the trip error as a measure of trip preservation. It aims to evaluate how the correlation between the starting and ending domain cells of original trajectories is preserved. We compute the trip error as the Jensen–Shannon divergence between the empirical trip distribution (the distribution of all possible pairs of starting and ending domain cells) of $\mathcal{D}$ and that of $\hat{\mathcal{D}}$.

6.2.5 Length error

The length of a trajectory is defined to be the total number of points it contains. We use the length error as a measure of trajectory length preservation. It aims to evaluate how the number of points of original trajectories is preserved. We compute the trajectory length error as the Jensen–Shannon divergence between the empirical length distribution of original trajectories in $\mathcal{D}$ and that of synthetic trajectories in $\hat{\mathcal{D}}$.

6.2.6 Diameter error

The diameter of a trajectory is defined to be the maximum distance between any pair of its points [15,16]. We use the diameter error as a measure of trajectory diameter preservation. For this purpose, we first find the maximum diameter in $\mathcal{D}$ and quantize it into 20 equal-width buckets. Then, we compute the diameter error as the Jensen–Shannon divergence between the bucketized diameter distribution of original trajectories in $\mathcal{D}$ and that of synthetic trajectories in $\hat{\mathcal{D}}$.

6.2.7 Total distance error

We use the total distance error as a measure of trajectory traveled distance preservation. For this purpose, we first sum up the distance between consecutive points of each trajectory in $\mathcal{D}$ to calculate the total distance traveled by that trajectory. Then, we find the maximum traveled distance of all the trajectories and quantize it into 20 equal-width buckets. Subsequently, we compute the total distance error as the Jensen–Shannon divergence between the bucketized traveled distance distribution of original trajectories in $\mathcal{D}$ and that of synthetic trajectories in $\hat{\mathcal{D}}$.

6.3 Experimental results

In the following, we evaluate the impact of the maximum height of noisy cost-sensitive path trees, $h_{\max}$, on the performance of DP-MODR, when the total privacy budget ε is varied.

We first evaluate the count query error measure. For this purpose, we construct five different count query sets on each moving objects dataset, each one having a different maximum query size (namely 4, 8, 12, 16, and 20) and containing 10,000 randomly generated count queries. Each location in a count query is uniformly drawn from the set of domain cells of the underlying discretized spatial domain. Tables 4, 5 and 6 compare the average count query error of DP-MODR for different count query sets on Geolife, Taxi, and Porto under different values of ε and $h_{\max}$. From the tables, we observe that when $h_{\max}$ is set to 3, DP-MODR generally produces lower count query error than when $h_{\max}$ is set to 2 or 4, especially for Geolife and Porto. Although DP-MODR yields slightly better results for Taxi when $h_{\max}$ is set to 2, its results are also good when $h_{\max}$ is set to 3. We also observe that the average count query error of the coarse-grained scenario is higher than that of the fine-grained one. The reason is that in the coarse-grained scenario, count queries are made on larger domain cells, and thus, the errors applied to small domain cells when constructing the synthetic moving objects database are aggregated.

Then, we evaluate other measures. It is worth mentioning that the locations rank correlation and frequent patterns rank

Table 4 Average count query error of DP-MODR for different count query sets on Geolife under different values of ε and $h_{\max}$

Scenario	Maximum query size	$\varepsilon = 0.05$			$\varepsilon = 0.1$			$\varepsilon = 0.5$		
		$h_{\max}$			$h_{\max}$			$h_{\max}$		
		2	3	4	2	3	4	2	3	4
Fine-grained	4	197.71	15.52	17.78	15.15	13.74	14.92	11.68	11.76	12.17
	8	101.58	7.56	9.11	7.74	6.50	7.74	5.67	5.71	6.03
	12	66.39	5.25	6.38	5.00	4.52	4.75	3.88	3.90	4.12
	16	66.54	4.05	4.06	3.94	3.45	3.63	2.99	3.03	3.15
	20	31.83	3.03	3.09	2.73	2.41	2.87	2.03	2.05	2.17
Coarse-grained	4	2111.32	62.97	91.14	56.46	39.95	57.78	22.25	22.79	28.79
	8	1053.30	33.18	48.02	30.01	20.35	29.66	11.30	11.56	14.77
	12	517.80	19.79	26.91	18.30	12.77	17.17	7.57	7.73	9.43
	16	558.00	15.42	24.72	14.98	9.90	13.97	5.70	5.85	7.38
	20	345.44	11.13	16.24	11.73	7.50	10.81	4.20	4.34	5.48

Table 5 Average count query error of DP-MODR for different count query sets on Taxi under different values of ε and $h_{\max}$

Scenario	Maximum query size	$\varepsilon = 0.05$			$\varepsilon = 0.1$			$\varepsilon = 0.5$		
		$h_{\max}$			$h_{\max}$			$h_{\max}$		
		2	3	4	2	3	4	2	3	4
Fine-grained	4	7.02	7.77	8.04	6.91	7.27	7.81	5.95	6.41	7.10
	8	3.45	3.79	3.97	3.34	3.55	3.85	2.75	3.03	3.43
	12	2.39	2.60	2.70	2.31	2.42	2.64	1.96	2.13	2.39
	16	1.83	2.05	2.12	1.81	1.91	2.06	1.54	1.69	1.88
	20	1.19	1.36	1.43	1.18	1.24	1.37	0.96	1.07	1.24
Coarse-grained	4	14.23	17.73	18.90	13.43	15.55	17.81	16.46	14.80	13.62
	8	6.93	8.94	9.40	6.61	7.69	8.86	8.25	7.36	6.64
	12	4.48	5.64	5.92	4.31	4.92	5.66	5.31	4.79	4.39
	16	3.50	4.22	4.49	3.29	3.73	4.24	3.97	3.61	3.34
	20	2.57	3.37	3.51	2.47	2.83	3.30	3.03	2.74	2.48

Table 6 Average count query error of DP-MODR for different count query sets on Porto under different values of ε and $h_{\max}$

Scenario	Maximum query size	$\varepsilon = 0.05$			$\varepsilon = 0.1$			$\varepsilon = 0.5$		
		$h_{\max}$			$h_{\max}$			$h_{\max}$		
		2	3	4	2	3	4	2	3	4
Fine-grained	4	0.49	0.44	0.45	0.45	0.43	0.51	0.43	0.43	0.43
	8	0.25	0.22	0.23	0.24	0.22	0.23	0.22	0.22	0.22
	12	0.18	0.17	0.17	0.17	0.16	0.16	0.16	0.16	0.16
	16	0.09	0.08	0.08	0.08	0.07	0.08	0.07	0.07	0.07
	20	0.10	0.10	0.10	0.10	0.09	0.09	0.09	0.09	0.09
Coarse-grained	4	4.52	3.48	4.01	4.08	3.61	3.61	3.54	3.47	3.58
	8	2.18	1.70	1.92	1.97	1.73	1.72	1.71	1.67	1.73
	12	1.57	1.18	1.37	1.39	1.21	1.23	1.19	1.16	1.20
	16	1.23	1.00	1.12	1.15	1.02	1.02	1.01	0.99	1.02
	20	0.88	0.64	0.79	0.79	0.66	0.68	0.64	0.63	0.65

Table 7 Locations rank correlation, frequent patterns rank correlation, and trip error of DP-MODR for Geolife under different values of ε and $h_{\max}$

Scenario	Measure	$\varepsilon = 0.05$			$\varepsilon = 0.1$			$\varepsilon = 0.5$		
		$h_{\max}$			$h_{\max}$			$h_{\max}$		
		2	3	4	2	3	4	2	3	4
Fine-grained	Locations rank correlation	0.15	0.17	0.17	0.17	0.18	0.18	0.21	0.23	0.22
	Frequent patterns rank correlation	1.00	0.95	0.79	1.00	0.99	0.97	1.00	1.00	1.00
	Trip error	0.33	0.33	0.34	0.29	0.30	0.30	0.26	0.26	0.27
Coarse-grained	Locations rank correlation	0.35	0.47	0.45	0.39	0.51	0.42	0.68	0.70	0.53
	Frequent patterns rank correlation	0.36	0.38	0.21	0.40	0.40	0.37	0.41	0.41	0.39
	Trip error	0.16	0.16	0.25	0.12	0.12	0.15	0.12	0.12	0.11

Table 8 Locations rank correlation, frequent patterns rank correlation, and trip error of DP-MODR for Taxi under different values of ε and $h_{\max}$

Scenario	Measure	$\varepsilon = 0.05$			$\varepsilon = 0.1$			$\varepsilon = 0.5$		
		$h_{\max}$			$h_{\max}$			$h_{\max}$		
		2	3	4	2	3	4	2	3	4
Fine-grained	Locations rank correlation	0.64	0.65	0.65	0.67	0.67	0.67	0.70	0.70	0.71
	Frequent patterns rank correlation	0.45	0.43	0.46	0.47	0.44	0.39	0.46	0.46	0.44
	Trip error	0.48	0.49	0.50	0.48	0.48	0.49	0.47	0.47	0.47
Coarse-grained	Locations rank correlation	0.88	0.87	0.86	0.88	0.88	0.87	0.88	0.88	0.88
	Frequent patterns rank correlation	0.75	0.70	0.65	0.83	0.75	0.72	0.86	0.84	0.83
	Trip error	0.18	0.16	0.16	0.18	0.17	0.17	0.20	0.19	0.18

correlation measures take a value between −1 and 1, with values closer to 1 signifying better utility. Also, trip error, length error, diameter error, and total distance error take a value between 0 and 1, with lower values signifying better utility. To compute the frequent patterns rank correlation, we consider 50 top frequent patterns of real trajectories. Tables 7, 8, 9, 10, 11 and 12 report the obtained results for DP-MODR under different values of ε and $h_{\max}$. From the tables, we observe that in most cases when $h_{\max}$ is set to 3, DP-MODR produces better results than, or as good as, when $h_{\max}$ is set to 2 or 4. It should be noted that the length error, diameter error, and total distance error measures are not dependent on the discretization of the underlying continuous spatial domain (see the definitions of these measures for more information), and thus, they yield the same values for both fine-grained and coarse-grained scenarios.

Furthermore, we evaluate the effectiveness of DP-MODRT, which is an extended version of DP-MODR to support time-dependent moving objects databases. For this purpose, we use three time-dependent measures, namely time-dependent count query error, time-dependent trip error, and time-dependent semi-trip error, to evaluate the spatial and temporal utilities of the synthetic moving objects databases constructed by DP-MODRT. Note that a time-dependent count query is a count query whose locations are time-dependent, and a semi-trip is a trajectory with a particular starting point. Also, the time-dependent trip (or semi-trip) error is a measure of trip (or semi-trip) preservation at around the same time. Time-dependent count queries have many applications in the real world. For example, in city planning or taxi service prediction, it is so useful to know the number of moving objects that have been in a particular location at a particular time. Table 13 reports the average time-dependent count query error of DP-MODRT for different time-dependent count query sets on Geolife, Taxi, and Porto under $\varepsilon = 0.5$. From the table, we observe that DP-MODRT has been able to answer time-dependent count queries of different sizes with a relatively low error rate and, thus, can be used well in various applications where time-dependent count queries are their building blocks. We also observe that the average time-dependent count query error of the coarse-grained scenario is higher than that of the fine-grained one, which is an expected result, as already observed in our previous experiments. Table 14 reports the time-dependent trip and semi-trip errors of DP-MODRT for Geolife, Taxi, and Porto under $\varepsilon = 0.5$. From the table, we observe that in the coarse-grained scenario, the results are better than the fine-grained one. The reason is that in the coarse-grained scenario, the spatial domain is discretized into larger time-dependent domain

Table 9 Locations rank correlation, frequent patterns rank correlation, and trip error of DP-MODR for Porto under different values of ε and $h_{\max}$

Scenario	Measure	$\varepsilon = 0.05$			$\varepsilon = 0.1$			$\varepsilon = 0.5$		
		$h_{\max}$			$h_{\max}$			$h_{\max}$		
		2	3	4	2	3	4	2	3	4
Fine-grained	Locations rank correlation	0.04	0.03	0.04	0.04	0.04	0.04	0.05	0.05	0.05
	Frequent patterns rank correlation	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76
	Trip error	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Coarse-grained	Locations rank correlation	0.24	0.29	0.28	0.28	0.26	0.29	0.31	0.35	0.33
	Frequent patterns rank correlation	0.62	0.62	0.62	0.63	0.62	0.62	0.69	0.63	0.63
	Trip error	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

Table 10 Length error, diameter error, and total distance error of DP-MODR for Geolife under different values of ε and $h_{\max}$

Measure	$\varepsilon = 0.05$			$\varepsilon = 0.1$			$\varepsilon = 0.5$		
	$h_{\max}$			$h_{\max}$			$h_{\max}$		
	2	3	4	2	3	4	2	3	4
Length error	0.10	0.10	0.10	0.10	0.10	0.09	0.09	0.09	0.09
Diameter error	0.10	0.10	0.14	0.08	0.06	0.08	0.11	0.08	0.06
Total distance error	0.06	0.06	0.10	0.05	0.05	0.06	0.06	0.05	0.06

Table 11 Length error, diameter error, and total distance error of DP-MODR for Taxi under different values of ε and $h_{\max}$

Measure	$\varepsilon = 0.05$			$\varepsilon = 0.1$			$\varepsilon = 0.5$		
	$h_{\max}$			$h_{\max}$			$h_{\max}$		
	2	3	4	2	3	4	2	3	4
Length error	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Diameter error	0.20	0.16	0.14	0.21	0.19	0.17	0.24	0.24	0.22
Total distance error	0.01	0.00	0.00	0.01	0.01	0.00	0.03	0.03	0.02

Table 12 Length error, diameter error, and total distance error of DP-MODR for Porto under different values of ε and $h_{\max}$

Measure	$\varepsilon = 0.05$			$\varepsilon = 0.1$			$\varepsilon = 0.5$		
	$h_{\max}$			$h_{\max}$			$h_{\max}$		
	2	3	4	2	3	4	2	3	4
Length error	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13
Diameter error	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Total distance error	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

cells, and thus, the starting and ending points of trajectories are more likely to be preserved.

6.4 Attack resilience analysis

As we said before, differential privacy is provably resilient to known and unknown attacks [12] and satisfies a strong and quantifiable mathematical privacy guarantee [10]. In this section, for further analysis and discussion, we design an attack on synthetic moving objects databases, which we call-sensitive locations disclosure attack and show to what extent DP-MODR is resilient to it. A similar attack resilience analysis can be performed for DP-MODRT.

Let us assume that there are some sensitive regions or locations, such as airports or hospitals, in the underlying spatial domain. The adversary has some background knowledge about the trip of a particular moving object and wants to inform about its presence in the sensitive locations. Here, we assume that the adversary knows the starting and ending points of the trip of the moving object.

We quantitatively measure how much information the adversary can get about the presence of moving objects

Table 13 Average time-dependent count query error of DP-MODRT for different time-dependent count query sets on Geolife, Taxi, and Porto under $\varepsilon = 0.5$

Scenario	Maximum time-dependent query size	Geolife	Taxi	Porto
Fine-grained	4	7.33	10.97	0.28
	8	2.48	6.77	0.20
	12	1.25	5.46	0.05
	16	1.29	2.55	0.06
	20	0.60	2.03	0.07
Coarse-grained	4	19.94	17.54	1.20
	8	9.53	9.09	0.67
	12	6.55	5.77	0.28
	16	4.50	5.27	0.32
	20	4.46	3.29	0.21

Table 14 Time-dependent trip and semi-trip errors of DP-MODRT for Geolife, Taxi, and Porto under $\varepsilon = 0.5$

Scenario	Measure	Geolife	Taxi	Porto
Fine-grained	Time-dependent trip error	0.53	0.67	0.65
	Time-dependent semi-trip error	0.31	0.59	0.37
Coarse-grained	Time-dependent trip error	0.43	0.27	0.46
	Time-dependent semi-trip error	0.12	0.04	0.00

Table 15 Average SLSR of DP-MODR for different sensitive location sets on Geolife, Taxi, and Porto under different values of ε

Scenario	Percent of sensitive locations	Geolife			Taxi			Porto		
		ε			ε			ε		
		0.05	0.1	0.5	0.05	0.1	0.5	0.05	0.1	0.5
Fine-grained	1	0.02	0.05	0.03	0.03	0.03	0.03	0.00	0.00	0.00
	5	0.05	0.08	0.05	0.04	0.04	0.04	0.07	0.07	0.07
	10	0.07	0.08	0.06	0.04	0.04	0.04	0.07	0.07	0.07
Coarse-grained	1	0.08	0.08	0.07	0.19	0.20	0.26	0.00	0.00	0.00
	5	0.11	0.11	0.11	0.34	0.35	0.39	0.20	0.20	0.21
	10	0.26	0.27	0.26	0.37	0.39	0.42	0.20	0.21	0.22

in sensitive locations by launching the sensitive locations disclosure attack. For this purpose, we compute a similarity ratio, called sensitive locations similarity ratio (SLSR), which indicates how much the sensitive locations (points) of an original trajectory are similar to those of synthetic trajectories. Since there is no any specific synthetic trajectory for a particular original trajectory in a differentially private (synthetic) moving objects database, we first individually compute the Jaccard similarity coefficient between the sensitive locations of an original trajectory and those of each synthetic trajectory with the same starting and ending points as the starting and ending points of the original trajectory. We then consider the average of these similarity coefficients as the sensitive locations similarity ratio of the original trajectory. The SLSR measure takes a value between 0 and 1, with lower values indicating more resistant to the sensitive locations disclosure attack and, thus, stronger privacy guarantee.

In the following, we evaluate the resiliency of DP-MODR to the sensitive locations disclosure attack. For this purpose, we construct three different sensitive location sets over the discretized spatial domain, each one having a different number of sensitive locations (namely 1%, 5%, and 10% of the total number of domain cells of the underlying discretized spatial domain). Each location in a sensitive location set is uniformly drawn from the set of all domain cells. We compute the SLSR measure for each original trajectory and average among them. Table 15 compares the average SLSR of DP-MODR for different sensitive location sets on Geolife, Taxi, and Porto under different values of ε. Since the size of Taxi and Porto is large, we randomly select 10,000 original trajectories that passed through at least one sensitive location and average among their SLSR values. From the tables, we

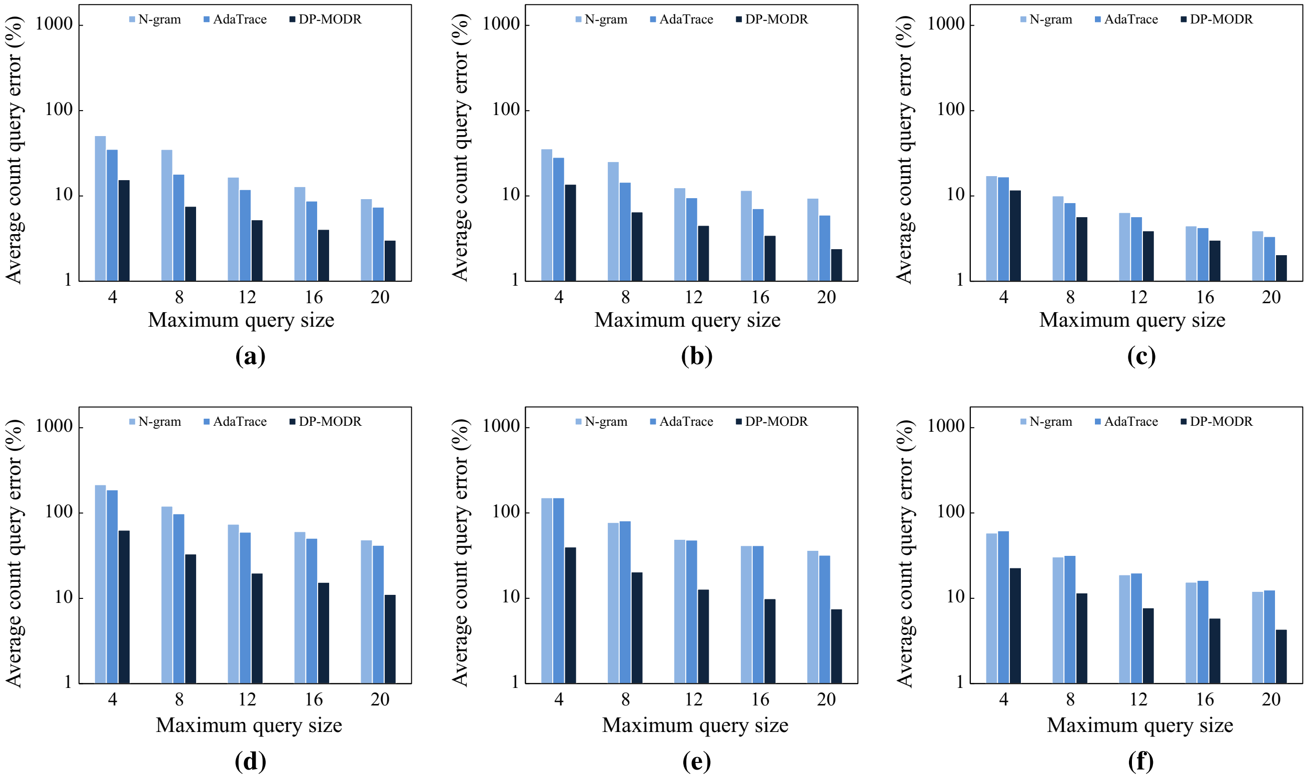

Fig. 5 Average count query error of N-gram, AdaTrace, and DP-MODR for different count query sets on Geolife under different values of ε: **a** Fine-grained, $\varepsilon = 0.05$. **b** Fine-grained, $\varepsilon = 0.1$. **c** Fine-grained, $\varepsilon = 0.5$. **d** Coarse-grained, $\varepsilon = 0.05$. **e** Coarse-grained, $\varepsilon = 0.1$. **f** Coarse-grained, $\varepsilon = 0.5$

observe that DP-MODR can well resist the sensitive locations disclosure attack. More specifically, in the fine-grained scenario, the results are better than the coarse-grained one. The reason is that in the coarse-grained scenario, the spatial domain is discretized into larger domain cells, and thus, the number of duplicate points in the moving objects database is much higher than the fine-grained one. Thus, in comparison with the fine-grained scenario, the synthetic trajectories are more similar to the original ones. We also observe that the average SLSR is smaller for the smaller sets of sensitive locations (smaller percent of sensitive locations). This is because as the size of sensitive location sets decreases, the likelihood that the sensitive points of an original trajectory are similar to those of the synthetic trajectories becomes less. Moreover, for the small values of ε, namely 0.05, the SLSR values are usually smaller because of a stronger privacy guarantee.

6.5 Comparison

In the following, we compare the spatial and temporal utilities of DP-MODR with those of N-gram [2] and AdaTrace [16]. Due to limited space, we only report the comparison results for two moving objects datasets, namely Geolife that includes a broad range of outdoor movements and Taxi that contains a large number of trajectories (refer to Sect. 6.1 for more information). For N-gram, we discretize the continuous spatial domain by partitioning it into an 8×8 grid for Geolife and a 32×32 grid for Taxi, which approximately yield better utilities than other resolutions in most evaluation measures. It should be noted that N-gram and AdaTrace assume that the points of trajectories have no time stamp attribute and, thus, are not comparable with DP-MODRT.

To begin with, we evaluate the count query error measure. Figures 5 and 6 compare the average count query error of DP-MODR with that of N-gram and AdaTrace for different count query sets on Geolife and Taxi under different values of ε. Here, we use log scale instead of linear scale to show the results due to the large differences. From the figures, we observe that the average count query error of DP-MODR is often better than that of N-gram and AdaTrace. The reason for the large count query error of N-gram, especially for small values of ε, is that N-gram constructs an exploration tree [2] to answer count queries. The exploration tree keeps the mobility patterns of original trajectories, but it divides the total privacy budget among its different levels. This results in high count query error for small values of ε (especially for datasets like Geolife where the real frequencies of trajectories are small as well). However, in some cases, when the fine-grained scenario is considered, N-gram can obtain good results for large moving objects datasets (where trajectories

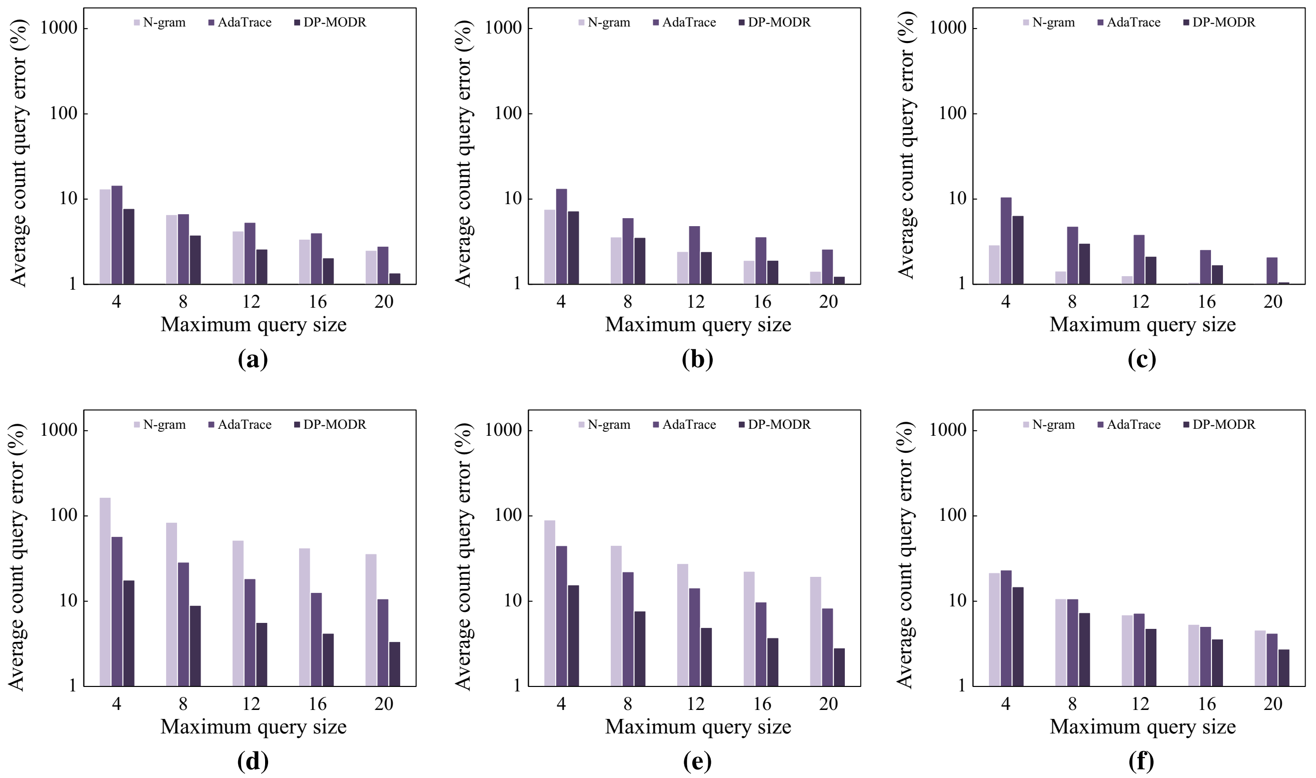

Fig. 6 Average count query error of N-gram, AdaTrace, and DP-MODR for different count query sets on Taxi under different values of ε: **a** Fine-grained, $\varepsilon = 0.05$. **b** Fine-grained, $\varepsilon = 0.1$. **c** Fine-grained, $\varepsilon = 0.5$. **d** Coarse-grained, $\varepsilon = 0.05$. **e** Coarse-grained, $\varepsilon = 0.1$. **f** Coarse-grained, $\varepsilon = 0.5$

usually have large frequencies) or large values of ε (where a small amount of noise is added to frequencies). This could be due to keeping the mobility patterns of original trajectories. However, two points need to be considered here: (1) N-gram incurs high time and space overheads, and (2) the importance of having good utility for large total privacy budgets is not as important as that for small total privacy budgets. Moreover, AdaTrace does not have good utility in count query answering as a result of the fact that it does not fully preserve the temporal relationships between points of each trajectory.

Then, we evaluate the locations rank correlation, frequent patterns rank correlation, and trip error measures. Tables 16 and 17 report the obtained results of N-gram, AdaTrace, and DP-MODR for Geolife and Taxi under different values of ε. Obviously, in most cases, DP-MODR has better results than N-gram and AdaTrace. This is because DP-MODR generates synthetic trajectories in a bottom-up way by iteratively joining the most probable paths, which allows it to preserve the mobility patterns and trajectory frequencies of the original moving objects database more efficiently.

Finally, we evaluate the length error, diameter error, and total distance error measures. Note that, as previously mentioned, the calculation of these measures does not depend on the resolution of the underlying discretized spatial domain, and thus, their values are the same for both fine-grained and coarse-grained scenarios. Figures 7 and 8 report the length error, diameter error, and total distance error of N-gram, AdaTrace, and DP-MODR for Geolife and Taxi under different values of ε. From the figures, we observe that DP-MODR achieves much less length error than N-gram and AdaTrace. The reason is that, by keeping the noisy median length of original trajectories starting in each domain cell, DP-MODR tries to preserve the distribution of trajectory lengths. It is worth mentioning that the distribution of trajectory lengths is one of the important properties of a moving objects database. However, as can be seen in the figures, none of N-gram and AdaTrace can preserve this distribution well. Moreover, the results of the diameter error and total distance error measures show that each of N-gram, AdaTrace, and DP-MODR performs better than the others in approximately 16.67%, 16.67%, and 66.67% of the cases, respectively. This shows that N-gram and AdaTrace could not preserve the distance traveled by original trajectories as well as DP-MODR.

7 Conclusion and discussion

Moving objects databases may contain detailed information about moving objects, and disclosing such information may reveal their preferences, lifestyles, social customs, and

Table 16 Locations rank correlation, frequent patterns rank correlation, and trip error of N-gram, AdaTrace, and DP-MODR for Geolife under different values of ε

Scenario	Measure	N-gram			AdaTrace			DP-MODR		
		ε			ε			ε		
		0.05	0.1	0.5	0.05	0.1	0.5	0.05	0.1	0.5
Fine-grained	Locations rank correlation	0.01	0.00	0.01	0.25	0.28	0.37	0.17	0.18	0.23
	Frequent patterns rank correlation	0.48	0.37	0.12	0.01	0.01	0.02	0.95	0.99	1.00
	Trip error	0.42	0.40	0.41	0.67	0.66	0.61	0.33	0.30	0.26
Coarse-grained	Locations rank correlation	0.35	0.33	0.49	0.44	0.50	0.59	0.47	0.51	0.70
	Frequent patterns rank correlation	0.34	0.46	0.06	0.03	0.03	0.02	0.38	0.40	0.41
	Trip error	0.28	0.28	0.24	0.52	0.48	0.34	0.16	0.12	0.12

Table 17 Locations rank correlation, frequent patterns rank correlation, and trip error of N-gram, AdaTrace, and DP-MODR for Taxi under different values of ε

Scenario	Measure	N-gram			AdaTrace			DP-MODR		
		ε			ε			ε		
		0.05	0.1	0.5	0.05	0.1	0.5	0.05	0.1	0.5
Fine-grained	Locations rank correlation	0.50	0.55	0.66	0.53	0.57	0.62	0.65	0.67	0.70
	Frequent patterns rank correlation	0.57	0.66	0.73	0.15	0.17	0.20	0.43	0.44	0.46
	Trip error	0.29	0.24	0.28	0.34	0.33	0.27	0.49	0.48	0.47
Coarse-grained	Locations rank correlation	0.72	0.76	0.85	0.66	0.73	0.83	0.87	0.88	0.88
	Frequent patterns rank correlation	0.68	0.71	0.62	0.37	0.36	0.35	0.70	0.75	0.84
	Trip error	0.08	0.07	0.08	0.21	0.19	0.10	0.16	0.17	0.19

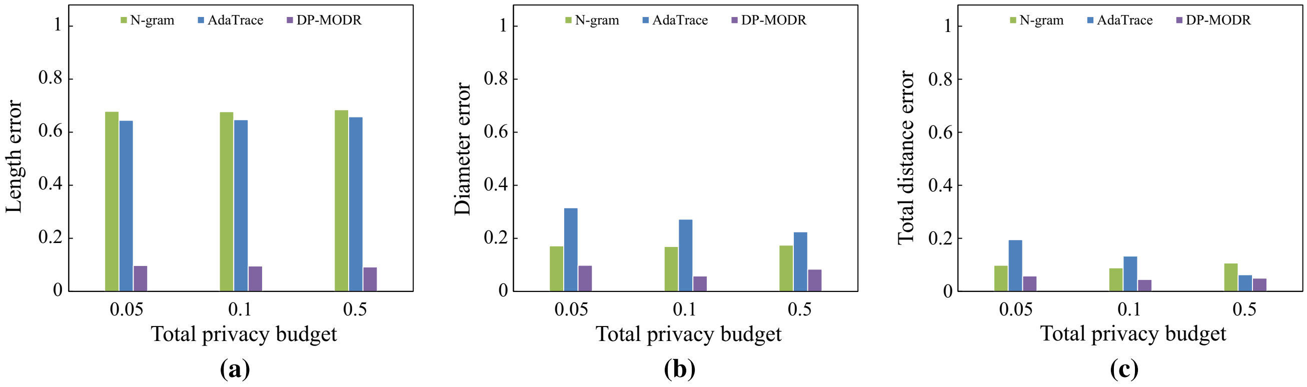

Fig. 7 Length error, diameter error, and total distance error of N-gram, AdaTrace, and DP-MODR for Geolife under different values of ε: **a** length error. **b** Diameter error. **c** Total distance error

sensitive personal information. For example, some potentially sensitive personal and professional information about a moving object can be obtained by knowing its presence at specific locations. Therefore, there is a growing concern about breaching the privacy of moving objects whose locations are monitored and tracked.

Differential privacy satisfies a strong and quantifiable mathematical privacy guarantee, which has provably resilient to known and unknown attacks. In particular, if a trajectory data release mechanism is differentially private, any information about a moving object is protected no matter what the adversary knows about that moving object. By the total privacy budget, the data owner can specify how to authorize certain analyses while being sure that certain privacy thresholds are not crossed. Therefore, any moving object can be assured that its presence in the database will most likely not reveal any information about it.

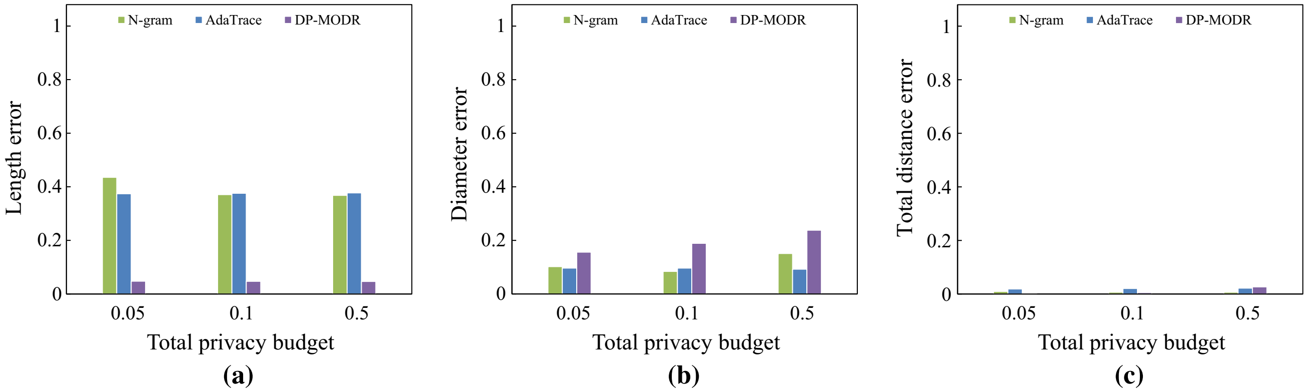

Fig. 8 Length error, diameter error, and total distance error of N-gram, AdaTrace, and DP-MODR for Taxi under different values of ε: **a** length error. **b** Diameter error. **c** Total distance error

Initially, differential privacy was designed to protect the output of statistical queries, and thus, releasing a differentially private version of an original database was not the aim of work on differential privacy. In recent years, some work has begun to use differential privacy for data release scenarios, which are more practical scenarios than previous ones. In this paper, we have continued this line of research by presenting DP-MODR, a differentially private mechanism for synthetic moving objects database release that tries to preserve spatial and temporal utilities as much as possible. DP-MODR first derives some useful properties of an original moving objects database, including the distribution of starting domain cells, the distribution of trajectory lengths, and mobility patterns, in a differentially private way. It then constructs some so-called noisy cost-sensitive path trees to keep existing most probable paths with different lengths (up to a maximum length) and different starting domain cells. It finally constructs a synthetic moving objects database by considering the obtained differentially private spatial and temporal properties of original trajectories. DP-MODR generates each synthetic trajectory in a bottom-up way by iteratively joining the most probable paths that are extracted from the constructed noisy cost-sensitive path trees.

In many practical location-aware applications, there is a serious need to release a strongly private version of an original moving objects database for effective data mining without learning any true information about moving objects. DP-MODR can be successfully used in such applications. For example, consider a hospital that uses an RFID patient tagging system in which patients' trajectories, personal data, and medical data are stored in a central moving objects database. The hospital intends to allow direct access to some parts of this database to data miners or movement pattern analysts for research purposes. Without having a robust mechanism to make strong guarantees about privacy, many patients will not be willing to share their information on the moving objects database. DP-MODR makes it possible to release synthetic trajectories while satisfying the strong guarantee of differential privacy and preserving the spatial and temporal utilities of original trajectories as well. Thus, DP-MODR assures patients that releasing the synthetic moving objects database will not reveal any information about them and also assures movement pattern analysts that the released differentially private moving objects database can be used for different data analysis tasks, including count query answering and frequent pattern mining.

Similar to most previous work, DP-MODR assumes that the points of trajectories have no time stamp attribute. However, in the real world, some queries may be time-dependent. An example of such queries in the real world is “number of trajectories that have the same starting and ending points at around the same time.” To address this problem, we have extended DP-MODR to preserve the time information of trajectories as well as the location information, in a differentially private way. In this extension, also known as DP-MODRT, in addition to discretizing the continuous spatial domain into a finite set of regions or domain cells (as we do in DP-MODR), we also discretize the time space into non-overlapping time intervals.

We have compared the spatial and temporal utilities of DP-MODR with those of N-gram [2] (a well-known differentially private mechanism for moving objects databases) and AdaTrace [16] (a recent differentially private mechanism for releasing trajectory data). Our comparison has shown that DP-MODR outperforms N-gram and AdaTrace by enhancing the utility of query answers and better preserves the main spatial and temporal properties of original trajectories. More specifically, in most cases, the average count query error of DP-MODR is better than that of N-gram and AdaTrace. This is due to the fact that DP-MODR tries to preserve the mobility patterns of original trajectories when constructing a synthetic moving objects database. In contrast, (1) N-gram constructs an exploration tree on an original moving objects database and, thus, has more time and space complexities and more noise error (especially for moving objects databases where the real frequencies of trajectories are small), and (2) AdaTrace removes some points of trajectories and, thus, does not fully preserve the temporal relationships between points of each trajectory. On the other hand, by multiple evaluation measures, namely locations rank correlation, frequent patterns rank correlation, trip error, length error, diameter error, and total distance error, we have shown that, in about 68.52% of the cases, DP-MODR can obtain better results than N-gram and AdaTrace. Moreover, we have used three time-dependent measures, namely time-dependent count query error, time-dependent trip error, and time-dependent semi-trip error, to evaluate the spatial and temporal utilities of the synthetic moving objects databases constructed by DP-MODRT. In fact, by these measures, we have tried to show how DP-MODRT preserves the time-dependent mobility patterns of trajectories.

DP-MODR can be successfully applied to any type of sequential databases where data records are ordered lists of items. More specifically, if we consider a sequence instead of a trajectory and, as a result, a sequential database instead of a moving objects database, DP-MODR can also be applied to this sequential database. Sequential databases are being increasingly used in a variety of analytical and commercial applications, such as web usage analysis and recommendation systems. The release of such databases is of vital importance to the advancement of these applications. As a typical scenario, consider a sequential database containing user journeys on a website. More specifically, each data record in this database is a time-ordered sequence of URL categories browsed by a user on the website. To release a differentially private version of this sequential database using DP-MODR, we first derive the distribution of sequence heads (the head of a sequence is considered to be its first item) and the distribution of sequence lengths, in a differentially private way. Then, we obtain the patterns among the items of sequences (URL categories) and the most probable paths among them while satisfying differential privacy. Finally, we construct a synthetic sequential database by generating synthetic sequences.

There is little work [14,31] that has shown that the Laplace distribution is not the optimal noise distribution to achieve differential privacy. For example, Soria-Comas et al. [31] built another distribution, based on the Laplace distribution, that still fulfils the conditions of differential privacy, and its probability mass is more concentrated toward zero. Therefore, as future work, we plan to achieve optimal differential privacy in our work by applying the optimal noise distribution instead of the Laplace distribution. Also, we plan to present differentially private mechanisms for other types of databases than moving objects databases, to efficiently release synthetic databases with strong privacy guarantees. Further, differential privacy guarantees that the adversary cannot distinguish pairs of synthetic databases based on slightly different original databases with a probability proportional to the total privacy budget. However, a more detailed and formal analysis of the extent to which differentially private mechanisms are resistant to such attacks can be considered as future work.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval The authors used three public anonymized moving objects datasets in which data records cannot be associated with any particular individual. So all procedures performed in studies involving human participants were in accordance with the ethical standards, as mentioned in the Menlo Report.

References

1. Al-Hussaeni, K., Fung, B.C.M., Iqbal, F., Liu, J., Hung, P.C.K.: Differentially private multidimensional data publishing. Knowl. Inf. Syst. **56**(3), 717–752 (2018). https://doi.org/10.1007/s10115-017-1132-3
2. Chen, R., Acs, G., Castelluccia, C.: Differentially private sequential data publication via variable-length n-grams. In: Proceedings of the 2012 ACM SIGSAC Conference on Computer and Communications Security. ACM, New York, NY, USA, pp. 638–649 (2012). https://doi.org/10.1145/2382196.2382263
3. Chen, R., Fung, B.C.M., Desai, B.C., Sossou, N.M.: Differentially private transit data publication: a case study on the Montreal transportation system. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York, NY, USA, pp. 213–221 (2012). https://doi.org/10.1145/2339530.2339564
4. Cormode, G., Jha, S., Kulkarni, T., Li, N., Srivastava, D., Wang, T.: Privacy at scale: local differential privacy in practice. In: Proceedings of the 2018 ACM SIGMOD International Conference on Management of Data. ACM, New York, NY, USA, pp. 1655–1658 (2018). https://doi.org/10.1145/3183713.3197390
5. Cormode, G., Kulkarni, T., Srivastava, D.: Answering range queries under local differential privacy. Proc. VLDB Endow. **12**(10), 1126–1138 (2019). https://doi.org/10.14778/3339490.3339496
6. Deldar, F., Abadi, M.: Differentially private count queries over personalized-location trajectory databases. Data Brief **20**, 1510–1514 (2018). https://doi.org/10.1016/j.dib.2018.08.104
7. Deldar, F., Abadi, M.: PLDP-TD: personalized-location differentially private data analysis on trajectory databases. Pervasive Mob. Comput. **49**, 1–22 (2018). https://doi.org/10.1016/j.pmcj.2018.06.005
8. Deldar, F., Abadi, M.: PDP-SAG: personalized privacy protection in moving objects databases by combining differential privacy and sensitive attribute generalization. IEEE Access **7**, 85887–85902 (2019). https://doi.org/10.1109/ACCESS.2019.2925236
9. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) Automata, Languages and Programming, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Germany, pp. 1–12 (2006). https://doi.org/10.1007/11787006_1
10. Dwork, C.: Differential privacy. In: van Tilborg, H.C.A., Jajodia, S. (eds.) Encyclopedia of Cryptography and Security. Springer US,

Boston, MA, USA, pp. 338–340 (2011). https://doi.org/10.1007/978-1-4419-5906-5_752
11. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) Theory of Cryptography, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Germany, pp. 265–284 (2006). https://doi.org/10.1007/11681878_14
12. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014). https://doi.org/10.1561/0400000042
13. Fung, B.C.M., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: a survey of recent developments. ACM Comput. Surv. **42**(4), 14:1–14:53 (2010). https://doi.org/10.1145/1749603.1749605
14. Geng, Q., Viswanath, P.: The optimal noise-adding mechanism in differential privacy. IEEE Trans. Inf. Theory **62**(2), 925–951 (2016). https://doi.org/10.1109/TIT.2015.2504967
15. Gursoy, M.E., Liu, L., Truex, S., Yu, L.: Differentially private and utility preserving publication of trajectory data. IEEE Trans. Mob. Comput. **18**(10), 2315–2329 (2019). https://doi.org/10.1109/TMC.2018.2874008
16. Gursoy, M.E., Liu, L., Truex, S., Yu, L., Wei, W.: Utility-aware synthesis of differentially private and attack-resilient location traces. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. ACM, New York, NY, USA, pp. 196–211 (2018). https://doi.org/10.1145/3243734.3243741
17. He, X., Cormode, G., Machanavajjhala, A., Procopiuc, C.M., Srivastava, D.: DPT: differentially private trajectory synthesis using hierarchical reference systems. Proc. VLDB Endow. **8**(11), 1154–1165 (2015). https://doi.org/10.14778/2809974.2809978
18. Holohan, N., Leith, D.J., Mason, O.: Differential privacy in metric spaces: numerical, categorical and functional data under the one roof. Inf. Sci. **305**, 256–268 (2015). https://doi.org/10.1016/j.ins.2015.01.021
19. Hou, J., Li, Q., Meng, S., Ni, Z., Chen, Y., Liu, Y.: DPRF: a differential privacy protection random forest. IEEE Access **7**, 130707–130720 (2019). https://doi.org/10.1109/ACCESS.2019.2939891
20. Jorgensen, Z., Yu, T., Cormode, G.: Conservative or liberal? Personalized differential privacy. In: Proceedings of the 2015 IEEE 31st International Conference on Data Engineering. IEEE Computer Society, Washington, DC, USA, pp. 1023–1034 (2015). https://doi.org/10.1109/ICDE.2015.7113353
21. Kartal, H.B., Liu, X., Li, X.B.: Differential privacy for the vast majority. ACM Trans. Manag. Inf. Syst. **10**(2), 8:1–8:15 (2019). https://doi.org/10.1145/3329717
22. Kohli, N., Laskowski, P.: Epsilon voting: mechanism design for parameter selection in differential privacy. In: Proceedings of the 2018 IEEE Symposium on Privacy-Aware Computing. IEEE, Piscataway, NJ, USA, pp. 19–30 (2018). https://doi.org/10.1109/PAC.2018.00009
23. Li, M., Zhu, L., Zhang, Z., Xu, R.: Achieving differential privacy of trajectory data publishing in participatory sensing. Inf. Sci. **400**, 1–13 (2017). https://doi.org/10.1016/j.ins.2017.03.015
24. Liu, C., Chakraborty, S., Mittal, P.: Dependence makes you vulnerable: differential privacy under dependent tuples. In: Proceedings of the 23rd Network and Distributed System Security Symposium. Internet Society, Reston, VA, USA, pp. 1–15 (2016). https://doi.org/10.14722/ndss.2016.23279
25. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: Proceedings of the 2007 48th Annual IEEE Symposium on Foundations of Computer Science. IEEE Computer Society, Washington, DC, USA, pp. 94–103 (2007). https://doi.org/10.1109/FOCS.2007.66
26. McSherry, F.D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data. ACM, New York, NY, USA, pp. 19–30 (2009). https://doi.org/10.1145/1559845.1559850
27. Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L.: Predicting taxi-passenger demand using streaming data. IEEE Trans. Intell. Transp. Syst. **14**(3), 1393–1402 (2013). https://doi.org/10.1109/TITS.2013.2262376
28. Niknami, N., Abadi, M., Deldar, F.: SpatialPDP: a personalized differentially private mechanism for range counting queries over spatial databases. In: Proceedings of the 2014 4th International Conference on Computer and Knowledge Engineering. IEEE, Piscataway, NJ, USA, pp. 709–715 (2014). https://doi.org/10.1109/ICCKE.2014.6993414
29. Piao, C., Shi, Y., Yan, J., Zhang, C., Liu, L.: Privacy-preserving governmental data publishing: a fog-computing-based differential privacy approach. Future Gener. Comput. Syst. **90**, 158–174 (2019). https://doi.org/10.1016/j.future.2018.07.038
30. Qardaji, W., Yang, W., Li, N.: Differentially private grids for geospatial data. In: Proceedings of the 2013 IEEE 29th International Conference on Data Engineering. IEEE Computer Society, Washington, DC, pp. 757–768 (2013). https://doi.org/10.1109/ICDE.2013.6544872
31. Soria-Comas, J., Domingo-Ferrer, J.: Optimal data-independent noise for differential privacy. Inf. Sci. **250**, 200–214 (2013). https://doi.org/10.1016/j.ins.2013.07.004
32. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. Int. J. Uncertain. Fuzziness Knowl. Based Syst. **10**(5), 571–588 (2002). https://doi.org/10.1142/S021848850200165X
33. Wang, S., Sinnott, R., Nepal, S.: Privacy-protected statistics publication over social media user trajectory streams. Future Gener. Comput. Syst. **87**, 792–802 (2018). https://doi.org/10.1016/j.future.2017.08.002
34. Wang, S., Sinnott, R.O.: Protecting personal trajectories of social media users through differential privacy. Comput. Secur. **67**, 142–163 (2017). https://doi.org/10.1016/j.cose.2017.02.002
35. Xu, C., Ren, J., Zhang, Y., Qin, Z., Ren, K.: DPPro: differentially private high-dimensional data release via random projection. IEEE Trans. Inf. Forensics Secur. **12**(12), 3081–3093 (2017). https://doi.org/10.1109/TIFS.2017.2737966
36. Xu, C., Zhu, L., Liu, Y., Guan, J., Yu, S.: DP-LTOD: differential privacy latent trajectory community discovering services over location-based social networks. IEEE Trans. Serv. Comput. (2018). https://doi.org/10.1109/TSC.2018.2855740
37. Zhang, J., Xiao, X., Xie, X.: PrivTree: a differentially private algorithm for hierarchical decompositions. In: Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data. ACM, New York, NY, USA, pp. 155–170 (2016). https://doi.org/10.1145/2882903.2882928
38. Zheng, Y., Zhang, L., Xie, X., Ma, W.Y.: Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th International Conference on World Wide Web. ACM, New York, NY, USA, pp. 791–800 (2009). https://doi.org/10.1145/1526709.1526816
39. Zheng, Z., Wang, T., Wen, J., Mumtaz, S., Bashir, A.K., Chauhdary, S.H.: Differentially private high-dimensional data publication in Internet of Things. IEEE Internet Things J. **7**(4), 2640–2650 (2020). https://doi.org/10.1109/JIOT.2019.2955503
40. Zhu, T., Li, G., Zhou, W., Yu, P.S.: Differentially private data publishing and analysis: a survey. IEEE Trans. Knowl. Data Eng. **29**(8), 1619–1638 (2017). https://doi.org/10.1109/TKDE.2017.2697856

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.