A Solution to Privacy Preservation in Publishing Human Trajectories

Xianming Li[1] and Guangzhong Sun[2*]
[1] School of Computer Science and Technology, University of Science and Technology of China
Hefei, China
[e-mail: lxm928@mail.ustc.edu.cn]
[2] School of Computer Science and Technology, University of Science and Technology of China
Hefei, China
[e-mail: gzsun@ustc.edu.cn]
*Corresponding author: Guangzhong Sun

Received July 9, 2017; revised June 24, 2019; accepted August 15, 2019; published August 31, 2020

Abstract

With rapid development of ubiquitous computing and location-based services (LBSs), human trajectory data and associated activities are increasingly easily recorded. Inappropriately publishing trajectory data may leak users' privacy. Therefore, we study publishing trajectory data while preserving privacy, denoted privacy-preserving activity trajectories publishing (PPATP). We propose S-PPATP to solve this problem. S-PPATP comprises three steps: modeling, algorithm design and algorithm adjustment. During modeling, two user models describe users' behaviors: one based on a Markov chain and the other based on the hidden Markov model. We assume a potential adversary who intends to infer users' privacy, defined as a set of sensitive information. An adversary model is then proposed to define the adversary's background knowledge and inference method. Additionally, privacy requirements and a data quality metric are defined for assessment. During algorithm design, we propose two publishing algorithms corresponding to the user models and prove that both algorithms satisfy the privacy requirement. Then, we perform a comparative analysis on utility, efficiency and speedup techniques. Finally, we evaluate our algorithms through experiments on several datasets. The experiment results verify that our proposed algorithms preserve users' privay. We also test utility and discuss the privacy-utility tradeoff that real-world data publishers may face.

Keywords: Activity trajectory, privacy, user model, adversary model, utility

This work was supported by the National Natural Science Foundation of China (No. 61303047) and the Youth Innovation Promotion Association of CAS. The authors thank Dr. Yinghua Zhou and Dr. Yu Huang for their valuable discussions for this work.

http://doi.org/10.3837/tiis.2020.08.010 **ISSN : 1976-7277**

1. Introduction

The last few years have seen the rapid development of smartphones and emerging location-based services (LBSs). LBS providers can effectively obtain user locations by GPS sensors as well as users' activities by various sensors [1][2] and specific applications (microblogs, tweets, etc.). LBS providers collect a large quantity of data for further use, such as point of interest (POI) recommendations and advertisements. Additionally, people are paying increasing attention to the big data collected by traditional smart card systems, which are most widely used in campuses and transportation systems. The locations and implicit activity information in the data are helpful for various data mining tasks, such as analysis of lifestyles and some personalized services.

In the real world, human trajectory data and activity information collected by an organization needs to be published to other organizations for various reasons, such as scientific research and administrative regulations [3]. Since raw data usually contain individual sensitive information, publishing such data may result in privacy leakage, which has a negative effect on both data publishers and users. Ensuring that the published data remain useful in practice while protecting individual privacy is quite challenging and thus attracts increasing attention [4].

Trajectory data are quite different from relational data, as studied in privacy-preserving data publishing (PPDP) [5]. The dependence between consecutive records may be exploited by potential adversaries to infer users' privacy. Furthermore, the attached activities may be considered sensitive by users and they should also be considered in privacy preservation mechanisms [6]. Many studies have been conducted to study privacy-preserving trajectory data publishing [3][7], but these approaches do not consider activity information or ignore the dependence between data and therefore cannot be applied for publishing activity trajectories.

Therefore, we study the problem of privacy-preserving activity trajectories publishing (PPATP) and propose S-PPATP, a solution to PPATP. S-PPATP consists of three steps. First, we formulate the problem by making necessary assumptions and defining appropriate parameters. Second, we devise privacy checking algorithms to guarantee that the published data satisfy the privacy requirement and optimize the data quality. Finally, we adjust the algorithms to meet practical requirements. In summary, we make the following contributions:

- Study PPATP. The difference between PPATP and the previous data publishing problems lies in the fact that the data contain users' activity information. To solve this problem, we propose a three-step solution, S-PPATP, which involves modeling, algorithm design and algorithm adjustment.
- Formulate PPATP from the aspects of privacy requirements, user and adversary behavior modeling and data quality metrics. In user behavior modeling, we propose an extended topic model for parameter learning in the hidden Markov model (Section 3).
- Propose two data publishing algorithms (PAs), PA-Markov and PA-HMM, for different user models. We prove that the algorithms both satisfy privacy requirements and optimize utility to some extent. We show that both algorithms use polynomial running time. Then, we propose several techniques to speed up the algorithms for better practical use (Section 4).
- Evaluate PA-Markov and PA-HMM on simulated and real-world datasets. The results show that both algorithms preserve privacy. We also test utility and discuss the

privacy-utility tradeoff that data publishers may face in real-world scenarios (Section 5).

2. Problem Statement

Trajectories consist of a sequence of geospatial points with timestamps. However, people would like to conduct activities when staying at a place of interest. Here, we follow the semantics of activity defined in [8].

Definition 1 (Activity): An activity $\alpha \in \mathbb{A}$ represents a type of human action that an individual can take, such as working and eating. $\mathbb{A}$ is a finite set that contains all the activities that can be performed by the users.

Activity information is similar to location semantic information, which is studied in [6], since location semantic information indicates what a person does in a place. However, an activity is not limited to a location semantic. For example, a student may post a tweet at a restaurant in addition to eating. In other words, here, activity is a more general concept than location semantic information. Location, timestamp and corresponding activity together make up an event about a given user.

Definition 2 (Event): Event e is a triple that includes activity information as well as when and where the user performs this activity, i.e., $e = \langle \alpha, t, l \rangle$, where $\alpha \in \mathbb{A}, t \in \mathbb{T}, l \in \mathbb{L}, e \in \mathbb{E} = \mathbb{A} \times \mathbb{T} \times \mathbb{L}$, $\mathbb{A}, \mathbb{T}, \mathbb{L}$ are predefined vocabulary of activity, time and location.

The granularity of the timestamp should be specified (e.g. hourly or every half day), and its legal values make up a finite set. Location and activity should be also finite variables. An event acts as a record in the dataset which may contain many records for a given user.

Definition 3 (Activity Trajectory): An activity trajectory Γ is a chronological sequence of events on a user, i.e., $\Gamma = \{e_1, e_2, \cdots, e_n\}$, where it is subject to $e_1.t \leq e_2.t \leq \cdots \leq e_n.t$.

Tables 1 and **2** are examples of dataset and activity trajectories.

Table 1. A sample of datasets

UserID	Location	Time	Activity
001	hotel	2017/3/9 7:07:20	having breakfast
002	factory	2017/3/9 8:35:11	working
001	center avenue	2017/3/9 9:45:53	riding a bike
002	McDonald's	2017/3/9 11:45:15	having lunch
001	Pizza Hut	2017/3/9 12:05:24	having lunch
002	swimming pool	2017/3/9 14:45:30	swimming
001	hospital	2017/3/9 16:37:44	seeing a doctor
002	hotel	2017/3/9 17:56:33	having dinner

Table 2. Activity trajectories of user 001 and 002

User	Activity trajectories
001	(having breakfast, 7:07:20, hotel) → (riding a bike, 9:45:53, center avenue) → (having lunch, 12:05:24, Pizza Hut) → (seeing a doctor, 16:37:44, hospital)
002	(working, 8:35:11, factory) → (having lunch, 11:45:15, McDonald's) → (swimming, 14:45:30, swimming pool) → (having dinner, 17:56:33, hotel)

In the real world, activity trajectories collected by an organization need to be published to other organizations for various reasons. For example, the transit data collected by smart card automated fare collection systems in transportation systems may be shared due to administrative regulations or profit sharing [3]. A typical scenario for data publishing is

described in **Fig. 1**. The data publisher collects data from the data generators or users and releases the collected data to a data miner or the public (e.g., on the Internet) who is called the data recipient. The data recipient will then use the published data for scientific research or other purposes. For example, an LBS provider collects its application users' check-ins and releases them to a scientific research organization to improve the recommendation algorithm in their application. In this case, the LBS provider is the data publisher, the application users are the data generators, and the public is the data recipient.

Fig. 1. A typical scenario of data publishing

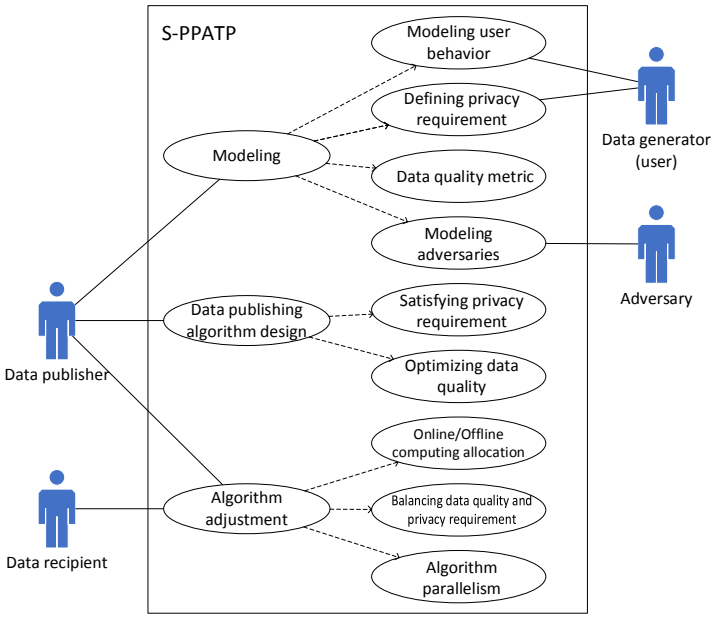


Fig. 2. Three tasks in PPATP

Activity trajectory data publishing benefits scientific research and helps to improve user experiences, but the individual sensitive information in raw data raises serious privacy concerns. For example, many people consider hospital as a sensitive location since it implies that he/she is ill. Additionally, sensitive information usually varies from person to person. For example, a hospital may be sensitive for a patient but may not be as sensitive to a doctor. Therefore, it is necessary to answer the following question: To preserve user privacy, what data can be published, and what data should be suppressed?

Fig. 2 illustrates the problem we want to solve and indicates the relationship between different tasks and roles. Overall, there are three tasks in PPATP. The first task helps to formulate the problem. Specifically, the following aspects should be covered.

- What is privacy, and to what extent should it be preserved (privacy requirement)?
- How powerful is the adversary (adversary model)? The adversary's background knowledge and reasoning methods have a substantial influence on the privacy preservation mechanism and thus should be well defined.
- Can user behaviors be modeled (user behavior modeling)? Everyone has his/her own living habits. If the adversary is a person who has knowledge of the user's habits (e.g., a friend or relative), he/she may utilize the behavior patterns to infer privacy.
- How should the published data be evaluated (data quality metric)? Activity trajectories are published for certain usage. To preserve user privacy, the published data will definitely be less useful than the raw data. It is important to consider to what extent the usefulness is affected, and therefore, a data quality metric is necessary.

The second task is designing a publishing algorithm based on the above modeling. The publishing algorithm should not only satisfy the privacy requirement but also ensure the data value. The last task is adjusting the publishing algorithm to meet practical needs, including how the publishing algorithm is implemented and how the parameters are selected.

We thus propose a solution to PPATP, S-PPATP, which is shown in **Fig. 3**. This solution consists of three steps corresponding to the three tasks in PPATP. First, we build models for the users and the adversary, define the privacy requirement and propose a utility function to measure the published data quality. Second, we propose two data publishing algorithms that satisfy the privacy requirement and optimize the utility function. Last, we conduct extensive experiments to elucidate how the publishing algorithms should be adjusted. We discuss the three steps in S-PPATP in detail in the following sections.

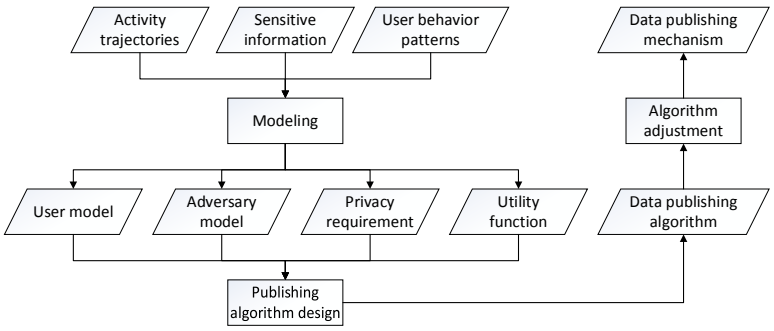


Fig. 3. Framework of S-PPATP

3. Modeling

3.1 Privacy requirement

Given user u with user model M and event space $\mathbb{E}$, the user is required to declare the information he/she does not want to publish. Sensitive information could be a location, a timestamp or an activity. For example, it is not good for an employee to let his/her boss know that he/she is on vacation because it implies that the employee is absent from work. In this case, the sensitive information to the employee is activity information (traveling) . Formally, a sensitive set $S \subset \mathbb{A} \cup \mathbb{T} \cup \mathbb{L}$ is identified for each user.

Informally, if the adversary gets no more knowledge about the sensitive information of u after accessing the published activity trajectory, we say it preserves the privacy of user u. Here, we apply the probabilistic attack model [5], which aims to achieve the uninformative principle [9]. Under this principle, the posterior probability of each type of sensitive information at

every sampling point is not much larger than the prior probability. Here, we apply $\delta\text{-}privacy$ [7] and extend its semantics by also regarding activity and timestamp as possible sensitive information.

Definition 4 (δ-privacy): For an publishing algorithm $\mathcal{A}$, it preserves $\delta\text{-}privacy$ if for all possible input activity trajectories generated from user model M, for all possible output O and all types of sensitive information $s \in S$

$$P[e_i.x = s|O] - P[e_i.x = s] \leq \delta \quad (1)$$

where $1 \leq i \leq n, x \in \{a, t, l\}$ is the type of s.

Publishing lgorithm $\mathcal{A}$ checks each event of the given user and determine whether to publish or suppress it. If the decision is "publish", $\mathcal{A}$ produces the event. If the decision is "suppress", $\mathcal{A}$ produces "NIL". In real-world applications, if the decision of an event is "NIL", the data should be replaced by "Unknown" or other default values.

We say the user's privacy is breached at position i of some output O if $\exists s \in S, P[e_i.x = s|O] - P[e_i.x = s] > \delta$, where x is an element of event. To measure the extent to which the privacy is breached in a dataset, we define $breach\ rate$. Assume that dataset D consists of data from U users. The breach rate refers to the ratio of events where the privacy is breached, i.e.,

$$Breach\ rate(D) = \frac{1}{U}\sum_u \frac{|\{i|P[e_i.x=s|D]-P[e_i.x=s]>\delta\}|}{n} \quad (2)$$

In the following, the breach rate of algorithm $\mathcal{A}$ refers to the breach rate of $\mathcal{A}$'s output dataset.

3.2 Data quality

Intuitively, if we always suppress the whole activity trajectory, the privacy would never be breached, which does not make sense since nothing is published. The more truthful data is published, the more useful the dataset is. We measure the quality of a dataset D by defining the utility as the expected fraction of truthful events in the output activity trajectory.

$$Utility(\mathrm{D}) = \mathrm{E}(\text{Fraction of truthful events}) \quad (3)$$

3.3 User model

Despite different lifestyles, a deep-rooted regularity is hidden behind human daily behaviors [10]. Since it has been proven that the Markov chain is useful in modeling human behavior patterns [11], we utilize two optional models to describe user behaviors: a Markov-based model and an HMM-based model.

3.3.1 Markov-based model

We use a Markov model to characterize individual behavior patterns. Specifically, we regard an activity trajectory Γ as three independent sequences: activity sequence (A), time sequence (T) and location sequence (L). The sequences are generated from Markov chains $M_a, M_t,$ and M_l, respectively. According to the property of Markov chain, the current state only depends on the the previous state, i.e.,

$$P[x\_i \,|\, x\_1, x\_2, \cdots, x\_(i-1)\,] = P[x\_i \,|\, x\_(i-1)\,], 2 \leq i \leq n, x \in \{a, t, l\} \quad (4)$$

3.3.2 HMM-based model

Diverse human behaviors sometimes share something in common. For example, some people go to a restaurant for breakfast, while others may go to the bakery or make other choices. Nevertheless, all of these actions imply the same topic: having breakfast. Such hidden topic usually cannot be explicitly observed from the data. The HMM is suitable for this situation

where a latent variable exists. A typical HMM model is defined by initial state distribution (π), transition probabilities (Tr) and emission probabilities (B). π and Tr determine the state sequence, which indicates individual behavior patterns and cannot be observed. B determines the observation sequence that corresponds to a user's activity trajectory.

In the HMM, the space of hidden states is usually much smaller than the space of observation variables. Therefore, dimension reduction must be leveraged for modeling. Here, we resort to topic modeling [12], which extracts a set of common "topics" from activity trajectories. Drawing an analogy, activity trajectories can be regarded as documents, and events can be regarded as words. Thus, topics can be explained as users' behavior patterns. For example, eating and entertainment after work could be two topics. In the former topic, words such as {8:00, having a bill, McDonald's} may appear. In the latter topic, words such as {18:40, watching TV, home} may appear. In other words, topic models reduce the complexity of activity trajectories by providing an interpretable low-dimensional representation.

A typical implementation of topic model is latent Dirichlet allocation (LDA). However, LDA does not consider word dependence . Although topic constraint of the same sentence can be added to LDA [13], the topics of different sentences are still independent so the method does not fit our situation. Therefore, we extend LDA to incorporate topic transition behind consecutive words to capture the temporal dependence between consecutive events which is common in daily lives. For instance, people usually start working after breakfast and watch TV after dinner. These correlations reflect people's regularity in daily life.

The samples can be generated by the following process:

1. For each document $u \in \{1,2,\cdots,U\}$ and topic $k \in \{1,2,\cdots,K\}$, draw $\theta_{u,k} \sim Dir(\alpha_k)$.
2. For each topic $k \in \{1,2,\cdots,K\}$, draw $\phi_{k,t} \sim Dir(\beta_t)$, $\phi_{k,a} \sim Dir(\beta_a)$, and $\phi_{k,l} \sim Dir(\beta_l)$.
3. For each word $w_i \in u$, draw topic $z_{u,i} \sim Mult\left(\theta_{u,z_{u,i-1}}\right)$, time $t_{u,i} \sim Mult\left(\phi_{z_{u,i},t}\right)$, activity $a_{u,i} \sim Mult\left(\phi_{z_{u,i},a}\right)$ and location $l_{u,i} \sim Mult(\phi_{z_{u,i},l})$.

$Mult$ and Dir represent multinomial distributions and Dirichlet distributions, respectively. θ is a distribution over topics for a document. ϕ_k is a discrete distribution over time, activity or location of topic k. α and β are hyperparameters for Dirichlet distributions.

For parameter learning, it is intractable to optimize the log-likelihood of observed samples, as analyzed in [12]. There are many approximation algorithms to solve this problem, such as variational expectation-maximization (EM) and Gibbs sampling. Due to the easier derivation of Gibbs sampling, we also use Gibbs sampling for approximation inference. Gibbs sampling is one method of Markov chain Monte Carlo, which approximates the posterior distribution over topic sequence $P(\mathbf{z}|a_{1:n},t_{1:n},l_{1:n})$, with an activity trajectory of length n. Gibbs sampling samples the topic z_i of a word i, i.e., the pair $\langle a_i,t_i,l_i\rangle$ conditioned on all the other words given and iteratively samples the topic of every word until convergence. Particularly, the following posterior probability is used to sample the topic z_i of the word :

$$\begin{aligned} P\left[z_i\middle|\mathbf{z}^{-i},a_{1:n},t_{1:n},l_{1:n}\right] &\propto P\left[z_i\middle|\mathbf{z}^{-i},\alpha\right]\cdot P\left[t_i\middle|z_i,t_{1:n}^{-i},\mathbf{z}^{-i},\beta_t\right] \\ \cdot\ P\left[a_i\middle|z_i,a_{1:n}^{-i},\mathbf{z}^{-i},\beta_a\right]&\cdot\ P\left[l_i\middle|z_i,l_{1:n}^{-i},\mathbf{z}^{-i},\beta_l\right] \end{aligned} \tag{5}$$

where superscript $-i$ means ignoring the ith token. Considering topic transition, the topic of the ith sample z_i depends on both the multinomial parameter and the previous topic z_{i-1}. Therefore, the first term on the right side of $\propto$ is:

$$P\left[z_i\middle|\mathbf{z}^{-i},\alpha\right] \propto \frac{n_{z_{i-1},z_i}^{u}+\alpha}{n_{z_{i-1}}^{u}+K\alpha}\frac{n_{z_i,z_{i+1}}^{u}+I(z_{i-1}=z_i=z_{i+1})+\alpha}{n_{z_i}^{u}+I(z_{i-1}=z_i)+K\alpha} \tag{6}$$

where K is the number of topics. The second term on the right side of $\propto$ in (6) adjusts the transition count from z_{i-1} to z_i since z_i is excluded. The other terms in (5) can be calculated using the following probabilities:

$$P\left[t_i \middle| z_i, t_{1:n}^{-i}, \mathbf{z}^{-i}, \beta_t\right] \propto \frac{n_{z_i,t_i}+\beta_t}{n_{z_i}+T\beta_t} \tag{7}$$

$$P\left[a_i \middle| z_i, a_{1:n}^{-i}, \mathbf{z}^{-i}, \beta_a\right] \propto \frac{n_{z_i,a_i}+\beta_a}{n_{z_i}+A\beta_a} \tag{8}$$

$$P[l\_i \mid z\_i, l\_(1{:}n)\text{^}(-i), \mathbf{z}\text{^}(-i), \beta\_l\,] \propto (n\_(z\_i, l\_i) + \beta\_l)/(n\_(z\_i) + L\beta\_l) \tag{9}$$

where T, A and L represent the vocabulary size of time, location and activity, respectively. The above sampling is performed iteratively until the sampling result change little. Then, we can estimate the document-specific topic distribution

$$p[k|u] = \frac{n_k^u+\alpha}{\sum_k(\, n_k^u+\alpha)} \tag{10}$$

and topic-specific vocabulary distribution

$$\phi_{k,w} = \frac{n_{\mathrm{k,w}}+\beta_{\mathrm{w}}}{\sum_{\mathrm{w}}(\, \mathrm{n_{k,w}}+\beta_w)}, w \in \langle t, a, l\rangle \tag{11}$$

and the topic transition probability of each user

$$p[z_i|z_{i-1}, u] = \frac{n_{z_{i-1},z_i}^u+\alpha}{n_{z_{i-1}}^u+K\alpha} \tag{12}$$

The output distributions of Gibbs sampling act as the three basic parameters of a HMM, i.e., state transition probability ($p[z_i|z_{i-1}, u]$), emission probability ($\phi_{k,w}$) and prior state distribution ($p[k|u]$) for all $1 \le i \le n, k \in \mathbb{E}, j \in \mathbb{E}$.

3.4 Adversary model

We study a powerful adversary who accesses the whole published dataset and owns two types of background knowledge. 1) User model. It means that the adversary knows well about the users' habits. 2) Publishing algorithm. Such an adversary knows when sensitive information is suppressed in the algorithm and can infer users' privacy from the additional information leaked by the suppression rules.

This type of adversary exists in our daily lives, e.g. our close friends, because an individual's behavior patterns can be easily learned by his/her close friends. Bayesian inference can be used by the adversary to infer sensitive information. Given a user model M, the adversary estimates the prior probability of the sensitive information at each position, i.e., $P[e_i.x = s](1 \le i \le n, x \in \{a, t, l\}, s \in S)$. Upon observing a published activity trajectory O, the adversary updates his/her inference by computing the posterior probability.

One technique an adversary can use to infer sensitive information is by event dependence. There are two types of dependence: $external$ dependence and $internal$ dependence, or the correlation between consecutive events and among the information inside an event. For example, if l is a sensitive location to user u where he/she usually goes after lunch on Monday, then it makes no sense to merely suppress "go to l on Monday afternoon" because this event can be inferred using the dependence from the lunch event. In this case, the adversary infers sensitive information by external dependence. For another example, assume user v usually goes to l' for drugs; then, merely suppressing the activity "taking drugs" makes no sense if the adversary knows the correlation between l' and drugs. In this case, the adversary infers sensitive information by internal dependence.

Another technique for inferring is usage of the privacy-preserving mechanism. Sometimes "suppression" implies "publishing". For instance, government officials are not allowed to enter casino for gambling. Consider a publishing algorithm tries to preserve this privacy by

suppressing the event if and only if it contains "gambling" activity. When the adversary observes an activity suppression, he/she will infer easily the privacy due to his/her knowledge of the suppression rule.

4. Privacy-preserving Algorithms for Publishing Activity Trajectories

4.1 Algorithm for the Markov-based model

The pseudocode of the publishing algorithm for the Markov-based model (PA-Markov) is shown in Algorithm 1. For the four input of PA-Markov, Γ can be extracted from the raw dataset. δ and S are provided by data publishers or users. M is learned from the user's historical records. PA-Markov finally outputs a modified activity trajectory O, which preserves δ-privacy. When PA-Markov is executed on all users, the whole dataset is published. Here, we assume that activity trajectory of each user is independent. In other words, the adversary cannot infer a user's privacy with the help of other users' activity trajectories.

At each position i, PA-Markov first checks external dependence by $externalCheck$ and then internal dependence by $internalCheck$. Procedures of $externalCheck$ and $internalCheck$ are shown in Algorithm 2 and 3. Given a position i and current temporary output O, $externalCheck$ outputs true if and only if for all the possible values at position i and all the possible values in S, publishing the information will not breach δ-privacy.

Probability estimation of sensitive information is the key steps of $externalCheck$. Prior probability can be computed by:

$$P[e_j.x = s] = \left(\vec{b}'T^{j-1}\right)_s \tag{13}$$

where $x \in \{a, t, l\}$ and $\vec{b}$ is the initial distribution of x, T is the one-step transition matrix of M, and subscript s means the position where the information is s.

For the posterior probability, $P[e_j.x = s|\langle O, y\rangle]$, we consider a temporary output $O = \langle o_1, o_2, \cdots, o_i\rangle$. Let j' be the last position before or at position j at which an event was published. Let j'' be the first position after position j at which an event was published. If no such position exists, then $j'' = n + 1$. It was proven in [14] that:

$$P\left[e_j.x = s\middle|\langle O, y\rangle\right] = P\left[e_j.x = s\middle|e_{j'}.x = o_{j'}, e_{j''}.x = o_{j''}\right] \tag{14}$$

Using Markov assumption of the activity trajectory, the probability in (14) is:

$$\begin{aligned} P_{post_e}\left[e_j.x = s\middle|\langle O, y\rangle\right] &= P\left[e_j.x = s\middle|e_{j'}.x = o_{j'}, e_{j''}.x = o_{j''}\right] \\ &= \frac{P\left[e_{j''}.x=o_{j''}\middle|e_j.x=s\right]P\left[e_j.x=s\middle|e_{j'}.x=o_{j'}\right]}{P\left[e_{j''}.x=o_{j''}\middle|e_{j'}.x=o_{j'}\right]} \end{aligned} \tag{15}$$

Here, $externalCheck$ checks complete value set in case of leaking additional information because the adversary knows the publishing algorithm. For example, if e_{n-1} is published but e_n is suppressed, the adversary can infer that e_n contains sensitive information based on (14). The complete value set at position i is determined by M, starting from the nearest position before i where the true data is published. Specifically, assume i' is the nearest position where the output≠NIL; we can get $P[x_i|x_{i'}]$ by $i - i'$ steps of transition.

After checking the external dependence of a given activity trajectory, PA-Markov start to check the internal dependence. Here, we resort to the concept of frequent pattern mining. We define the posterior probability of sensitive information s given another information y_0 as the confidence of rule: $e.y = y_0 \Rightarrow e.x = s$, i.e.,

$$P_{post_i}[e.x = s|e.y = y_0] = conf(e.y = y_0 \Rightarrow e.x = s) = \frac{|\{e|e.x=s, e.y=y_0\}|}{|\{e|e.y=y_0\}|} \tag{16}$$

As shown in Algorithm 3, given an event e, $internalCheck$ will check each component of e that is not NIL. During each check, $internalCheck$ iterates on all the sensitive information in S to compute the posterior probability. If for some $s \in S$, the posterior probability $P[e.x = s|e.y]$ is δ larger than the prior probability $P[e.x = s]$, then $e.y$ is suppressed, where $y \in \{a, t, l\}$. Thus, $internalCheck$ ensures that for all the sensitive information $s \in S$, all the posterior probability $P[e.x = s|\cdot]$ is no δ larger than the prior probability $P[e.x = s]$, which preserves δ-privacy. Additionally, publishing or suppressing data in $internalCheck$ does not breach the result of $externalCheck$ since we do not use the current event for external dependence check in $externalCheck$.

Proposition 1: $externalCheck$ preserves δ-privacy in Definition 4.

Proof: We consider that user u has a sensitive set S and $externalCheck$ produces O. At any position j, we consider two cases: 1) If $j'' \leq n$, $externalCheck$ ensures that the posterior probability is not δ larger than the prior probability, so it publishes $e_{j''}.x$. After e_j'' is published, whether the events after $e_j''.x$ in Γ are published or suppressed has no influence on the posterior probability according to (14). Thus, $externalCheck$ preserves δ-privacy until termination of the algorithm. 2) If $j'' = n + 1$, consider how $externalCheck$ works when publishing $e_{j'}.x$. $ExternalCheck$ deciding to publish $e_{j'}.x$ implies that the posterior probability of $e_j.x = s$ is not δ larger than the prior probability given $j'' = n + 1$ during the check.

Since both $externalCheck$ and $internalCheck$ preserve δ-privacy, we have Proposition 2.

Proposition 2: PA-Markov preserves δ-privacy in Definition 4.

Algorithm 1 Publishing Algorithm for Markov-based model

Input: privacy requirement δ, user model *M*, sensitive set *S*, activity trajectory Γ
Output: activity trajectory *O* which preserves δ-privacy
$O \leftarrow \emptyset$;
for *i* = 1 → *n* **do**
  $e \leftarrow$ the *i*th event in Γ; $e' \leftarrow e$;
  if externalCheck(δ, M_a, i, O_a, S) == false **then** $e'.a \leftarrow$ NIL;
  end if
  if externalCheck(δ, M_t, i, O_t, S) == false **then** $e'.t \leftarrow$ NIL;
  end if
  if externalCheck(δ, M_l, i, O_l, S) == false **then** $e'.l \leftarrow$ NIL;
  end if
  internalCheck(e'); $O \leftarrow <O, e'>$;
end for
Publish generated activity trajectory *O*;
return *O*;

Algorithm 2 externalCheck

Input: privacy requirement δ, Markov model *M*, position *i*, current output *O*, sensitive set *S*
Output: the *i*th information is published (true) or suppressed (false)
for each possible value y **do**
  for each $s \in S$ **do**
    for $j = 1 \rightarrow n$ **do**
      Compute $p_{prior} \leftarrow P[e_j.x = s]$ and $p_{post\_e} \leftarrow P[e_j.x = s| < O, y >]$;
      if $p_{post\_e} - p_{prior} > \delta$ **then**
        return false;
      end if
    end for

```
    end for
end for
return true;
```

Algorithm 3 internalCheck

Input: event e
Output:

```
run componentCheck on each component of e;
return;

procedure componentCheck(e, y)
if y ≠ NIL then
    for each s ∈ S do
        if p[e.x = s|y] − p[e.x = s] > δ then
            y ← NIL; break;
        end if
    end for
end if
return;
```

4.2 Algorithm for HMM-based model

We assign a probability p_i^j for each event i with which i is published at position j. With probability $1 - p_i^j$, event i is suppressed at position j. We further define a publishing vector $\boldsymbol{p}$ containing all the publishing probabilities. Given the publishing vector $\boldsymbol{p}$, our publishing algorithm outputs each event with its publishing probability in order. The pseudocode is shown in Algorithm 4. By defining the publishing probability, whether an output activity trajectory breaches δ −privacy can be checked using a publishing vector. The pseudocode is shown in Algorithm 5. If and only if δ −privacy is not breached at any position for all the sensitive information, the algorithm returns true.

In $privacyCheck$, calculations of the prior and posterior probability of sensitive information are also crucial points. Prior probability can be computed by the Markov property. Assume we want to compute the prior probability of some sensitive information s at position i. The prior distribution of e_i can be computed by $i - 1$ step transition probability, i.e.,

$$P[e_i = e] = \left(\pi Tr^{i-1} B\right)_e \tag{17}$$

Then, the prior probability of s can be computed by

$$P[e_i.x = s] = \sum_{e.x=s} \left(\pi Tr^{i-1} B\right)_e \tag{18}$$

Computing posterior probability is more complex. Since an event will not be partially published in PA-HMM, there is no need to check internal dependence in particular. As each event has a publishing probability, the adversary cannot infer sensitive information from the original HMM. From the adversary's point of view, the output is generated from a new HMM, i.e., $M' = (\pi', Tr', B')$. The initial state distribution and transition probabilities do not change, i.e., $\pi' = \pi, Tr' = Tr$. However, emission probabilities, B', are changed to

$$b_{i,j}'^k = \begin{cases} b_{i,j} p_j^k & j \in \mathbb{E} \\ 1 - \sum_l b_{i,l} p_l^k & j = NIL \end{cases}, 1 \le k \le n \tag{19}$$

According to (3) in [7], the distribution of latent state Y is

$$P[Y_i = y|O] = \frac{\alpha_y(i)\beta_y(i)}{\sum_{y'} \alpha_{y'}(i)\beta_{y'}(i)}, 1 \le y \le K \tag{20}$$

where

$$\alpha_y(i) = P[Y_i = y, o_1, o_2, \cdots, o_{i-1}],\ \beta_y(i) = P[o_i, o_{i+1}, \cdots, o_n | Y_i = y] \tag{21}$$

α and β can be obtained by the forward-backward algorithm. Thus, we have

$$\begin{aligned} P[e_i.x = s|O] &= \sum_y P[Y_i = y|O] P[e_i.x = s|Y_i = y] \\ &= \sum_y P[Y_i = y|O] \cdot \sum_e I(e.x = s) b'^i_{y,e} \end{aligned} \tag{22}$$

Algorithm 4 Publishing Algorithm for HMM-based model

Input: privacy requirement δ, user model M, sensitive set S, activity trajectory Γ
Output: activity trajectory O which preserves δ-privacy
1: $O \leftarrow \emptyset$; $\boldsymbol{p} \leftarrow$ vectorSearch(δ, M, S);
2: **for** $i = 1 \rightarrow n$ **do**
3: $e \leftarrow$ NIL; With probability $p^i_{e_i}$, $e \leftarrow e_i$; $O \leftarrow < O, e >$;
4: **end for**
5: Publish O;

Algorithm 5 privacyCheck

Input: privacy requirement δ, user model M, sensitive set S, publishing vector p
Output: true or false
1: **for** each $s \in S$ **do**
2: **for** $i = 1 \rightarrow n$ **do**
3: Compute prior probability $P[e_i.x = s]$;
4: **for** all the possible activity trajectory O **do**
5: Compute posterior probability $P[e_i.x = s|O]$;
6: **if** $P[e_i.x = s|O] - P[e_i.x = s] > \delta$ **then**
7: **return** false;
8: **end if**
9: **end for**
10: **end for**
11: **end for**
12: **return** true;

Obviously, if we increase the publishing probability, the utility will also increase. Our goal is to search the publishing vector that maximizes utility while preserving $\delta-$privacy. However, privacyCheck is neither convex nor concave. We observe that if we decrease the publishing probability, we can certainly improve privacy. We say vector **p** dominates **q** if for all $i, j, p_i^j \leq q_i^j$. Then, we make the following proposition.

Proposition 3: If **p** preserves δ-privacy, then so does any **q** dominated by **p**.

Proof: Our proof is quite similar to the proof of the monotonicity property of the probabilistic check in [7]. Consider two publishing vectors, **p** and **q**. **p** is larger by ϵ in exactly one dimension: $p_i^j = q_i^j + \epsilon$. Assume that **p** preserves δ-privacy. It was proven in [7] that for all outputs O and sensitive information $s \in S$, the maximum posterior probability of e_j, $P[e_j|O]$, does not increase when the publishing vector goes from **p** to **q**. Thus, we have

$$\begin{aligned} P_{\mathbf{q}}[e_j.x = s|O] - P[e_j.x = s] &= \sum_i I(i.x = s) P_{\mathbf{q}}[e_j = i|O] q_i^j - P[e_j.x = s] \\ &\leq \sum_i I(i.x = s) P_{\mathbf{p}}[e_j = i|O] p_i^j - P[e_j.x = s] \\ &\leq P_{\mathbf{p}}[e_j.x = s|O] - P[e_j.x = s] \leq \delta \end{aligned} \tag{23}$$

Since the gap between posterior and prior probability does not increase when the publishing vector goes from **p** to **q**, the privacy is preserved. In other words, privacy is an anti-monotone property.

The range of $\mathbf{p}$ is $[0,1]^{n|\mathbb{E}|}$, which contains infinite vectors. We discretize the space $[0,1]$ to $[0,0.1,\cdots,0.9,1]$ and use a greedy algorithm, ALGP [15], to optimize $\mathbf{p}$, as shown in Algorithm 6. We call a privacy-preserving publishing vector $\mathbf{p}$ an extreme point if increasing any dimension of $\mathbf{p}$ will breach privacy. The idea of $vectorSearch$ is seeking all the extreme points that are maintained in $MaxTrueSet$ by iteratively using binary search. Then, the publishing vector with the highest utility is chosen from $MaxTrueSet$ and returned.

$PrivacyCheck$ determines whether a publishing vector preserves privacy by checking δ-privacy. Thus, PA-HMM preserves δ-privacy, as shown in the following proposition.

Proposition 4: PA-HMM preserves δ-privacy in Definition 4.

Algorithm 6 vectorSearch

Input: privacy requirement δ, user model M, sensitive set S
Output: publishing vector $\boldsymbol{p}$

MaxTrueSet = $\emptyset$; *Candidate* = $\{(0,0,\cdots,0)\}$;
while *Candidate* $\neq \emptyset$ **do**
  $\boldsymbol{p} \leftarrow$ some point in *Candidate*;
  if privacyCheck($\delta, M, \boldsymbol{p}$) = true **then**
    for $i = 1 \rightarrow n|\mathbb{E}|$ **do**
      $low \leftarrow \boldsymbol{p}[i]$; $\boldsymbol{p}[i] = 1$;
      if privacyCheck($\delta, M, \boldsymbol{p}$) = true **then continue**;
      end if
      $high = 1$;
      while $high - low \geq 0.1$ **do**
        $mid \leftarrow (high + low)/2$; $\boldsymbol{p}[i] \leftarrow mid$;
        if privacyCheck($\delta, M, \boldsymbol{p}$) = true **then** $low = mid$;
        else $high = mid$;
        end if
      end while
    end for
  end if
  MaxTrueSet $\leftarrow$ *MaxTrueSet* $\cup \{\boldsymbol{p}\}$;
  $Cand_{new} \leftarrow \emptyset$;
  for all $\boldsymbol{p}' \in Candidate$ **do**
    if $\boldsymbol{p}$ dominates $\boldsymbol{p}'$ **then**
      for $i = 1 \rightarrow n|\mathbb{E}|$ **do**
        $\boldsymbol{p}'' \leftarrow \boldsymbol{p}'$; $\boldsymbol{p}''[i] \leftarrow \boldsymbol{p}[i] + 0.1$;
        if $\boldsymbol{p}''$ is valid **then**
          $Cand_{new} \leftarrow Cand_{new} \cup \boldsymbol{p}''$;
        end if
      end for
    end if
  end for
  $Candidate \leftarrow Cand_{new}$;
end while
Select $\boldsymbol{p} \in MaxTrueSet$p with the highest utility;
return $\boldsymbol{p}$;

4.3 Comparison of PA-Markov and PA-HMM

4.3.1 Utility

We have proven that privacy is anti-monotone and utility is monotone; thus, we use a greedy search algorithm, ALGP, to maximize the publishing vector p. We optimize the publishing vector of PA-HMM. While PA-Markov is locally optimal, i.e., if the current information is published despite externalCheck deciding to suppress it, then the privacy may be breached if the temporary sequence is exactly the same as the final output. Although internalCheck can guarantee publishing as much information as possible, the whole algorithm, PA-Markov, is not globally optimal in utility.

4.3.2 Efficiency

We denote that $m = max\{|A|, |T|, |L|\}$. The time complexity of computing prior and posterior probability of PA-Markov are $O(nm^3)$. Sensitive set S is given and we denote its size by $|S|$. Since the innermost iteration of $externalCheck$ runs $O(n^2m|S|)$ times, the total time complexity of PA-Markov is $O(n^3m^4|S|)$.

For PA-HMM, the running time of Gibbs sampling is $O(RKn)$, where R refers to the number of iterations. Since we exploit a greedy binary search to maximize p, the number of calls to privacyCheck in vectorSearch is $O(n|\mathbb{E}|log(d))$, where d is the number of intervals we divide $[0,1]$ into. The time complexity of prior and posterior probability estimation is $O(n|\mathbb{E}|K^2)$ and $O(n|\mathbb{E}|^2)$. In addition, privacyCheck iterates for $O(n|S|)$ times. Since K is usually considerably less than $|\mathbb{E}|$, the total running time of PA-HMM is $O(n^3|\mathbb{E}|^3|S|log(d))$.

4.3.3 Speedup

To speed up PA-Markov, we can further improve the procedures of dependence check:

a) At position i, $externalCheck$ checks privacy breach by computing p_{prior} and p_{post} on all the positions. Assume i' is the nearest position before i where the event is published, according to (15), the events after i' does not affect the posterior probability of $e_{i''}.x = s$ for all $i'' \leq i'$ and all s. Thus, Line 3 in $externalCheck$ could be improved by:

1: $i' \leftarrow$ the last position before i where $e_{i'}.x$ is published

2: **for** $j = i' + 1 \rightarrow n$ **do**

b) In $externalCheck$, each possible y and each sensitive information $s \in S$ are checked separately, therefore we can use data parallelism to further accelerate the process. Thus, Line 1 to Line 2 in $externalCheck$ can be improved by:

1: **for** each possible value y parallel **do**

2: **for** each $s \in S$ parallel **do**

To speed up PA-HMM, we can run $vectorSearch$ offline since it is independent of the input activity trajectory. Then, we just need to start at Line 3 in Algorithm 4 online. Thus, the online running time is reduced to $O(n)$, which is acceptable for real-time applications.

5. Evaluation

5.1 Datasets

A small-scale simulated dataset and two real-world datasets are built in our experiments. The simulated dataset (denoted by SD) is generated randomly, while the real-world datasets (accessible at https://github.com/PPATP/Campus-smart-card) are collected using the campus

smart card system of a university in China. In the system, hundreds of point of sale (POS) machines are set up where payment is needed (e.g., canteens, stores) or for check in/out (e.g., libraries, dormitory). When a student swipes smart card on POS machine, a record including timestamp, expense, location and some other metadata is saved temporarily in the POS machine and later uploaded to centralized database. We use these records as students' activity trajectories in the experiment. We choose two datasets. One is collected from September to December 2011 (denoted by D11), and the other is from September to December 2015 (denoted by D15). The statistics are shown in **Table 3**.

Table 3. Statistics of the datasets

	Number of activities	Number of time intervals	Number of locations	Average activity trajectory length	Number of users
SD	4	3	8	50	10
D11/D15	4	3	6	50	50

Table 4. Experimental results on the simulated dataset

delta	Breach rate						Utility	
	Markov-based model			HMM-based model				
	PA-Markov	noMask	nointernal	PA-HMM	noMask	sensitiveaMask	PA-Markov	PA-HMM
0.1	0	0.244	0.564	0	0.201	0	0.387	0.88
0.3	0	0.244	0.024	0	0.001	0	0.481	0.934
0.5	0	0.244	0	0	0	0	0.482	0.936
0.7	0	0.244	0	0	0	0	0.686	0.938
0.9	0	0	0	0	0	0	0.968	0.938

5.2 Baselines

a) noMask. NoMask publishes raw activity trajectory without any suppression. According to the adversary model, the adversary knows this mechanism. Therefore the posterior probability of s is:

$$P[e_i.x = s|O] = \begin{cases} 1 & o_i.x = s \\ 0 & o_i.x \neq s \end{cases}, 1 \leq i \leq n \tag{24}$$

b) sensitiveMask. SensitiveMask is a naive approach that suppresses an event when there is sensitive information in the event and publishes it if there is not. When a suppression occurs, the adversary knows that it has sensitive information, but does not know the exact event. Here, the HMM is also applied for updating adversary's posterior probabilities. What the adversary observes is a new HMM M''. The initial state distribution and transition probabilities of M'' remain the same, while the emission probabilities are changed to:

$$b_j(k) = P[o_i = k|e_i = j] = \begin{cases} 1 & k = \text{NIL and } \exists j.x \in S \text{ or } k = j \\ 0 & \text{others} \end{cases} \tag{25}$$

for all $1 \leq i \leq n, k \in \mathbb{E} \cup \{\text{NIL}\}, j \in \mathbb{E}$. The posterior probability can be computed by (20).

c) PA-Markov without an internal check. We use PA-Markov without an internal check (denoted as nointernal) to test whether internalCheck is necessary. NoInternal is the same as PA-Markov except for deleting Line 10 in Algorithm 1.

5.3 Results on the simulated dataset

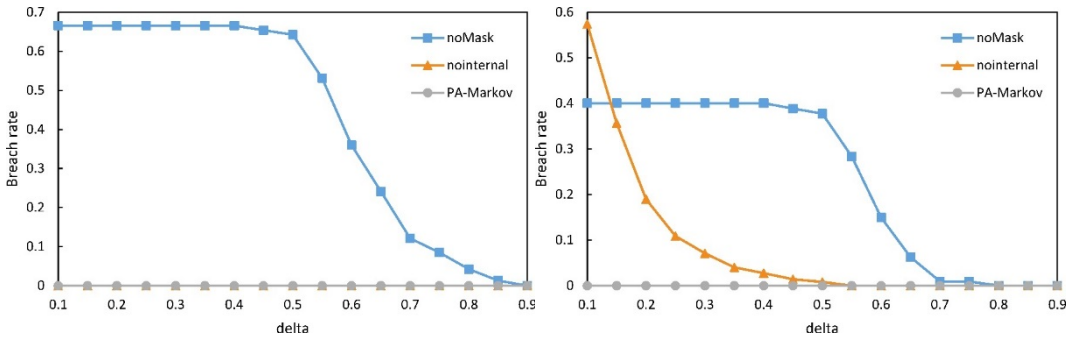
We randomly choose an activity as sensitive information ($|S| = 1$), and the experimental results on SD are shown in **Table 4**. Performing better than the baselines, the breach rates of PA-Markov and PA-HMM are always 0, which demonstrates that PA-Markov and PA-HMM preserve users' privacy. Additionally, the breach rate of sensitiveMask is also 0, which means

that merely suppressing sensitive information is sufficient to preserve the privacy of SD. In Section 5.4, we conduct a more detailed analysis of the experimental results on D11 and D15 and show that sensitiveMask sometimes causes breaches in users' privacy.

5.4 Results on the real-world datasets

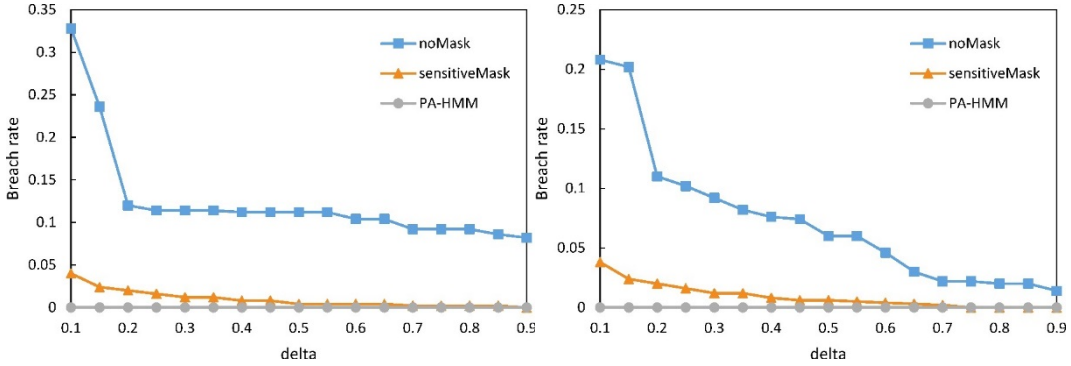
5.4.1 Privacy Breaches

The breach rates of PA-Markov and PA-HMM are shown in **Figs. 4, 5, 6** and **7**. We conducted experiments on two types of sensitive information: 1) a sensitive activity, a sensitive time stamp and a sensitive location 2) a sensitive activity or a sensitive time stamp or a sensitive location. All the sensitive sets are chosen randomly. NoMask has a high breach rate even when $\delta = 0.5$. When δ is small, noInternal has a very high breach rate, which indicates the risk of internal dependence attacks by the adversary. PA-Markov considers both external and internal dependence and therefore, always preserves privacy. For the HMM-based user model, we find that PA-HMM also performs the best. Simply suppressing all the events with sensitive information merely lowers the breach rate but cannot guarantee that all the privacy is preserved. As shown in **Figs. 5** and **7**, the adversary can sometimes still infer sensitive information using the correlation between events.



(a) Random a,t and l (b) Random a or t or l

Fig. 4. Breach rate of PA-Markov, noMask and nointernal on D11



(a) Random a,t and l (b) Random a or t or l

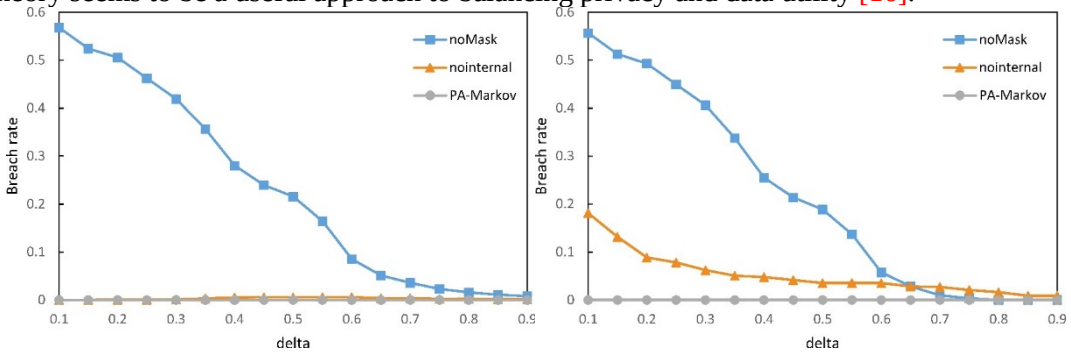
Fig. 5. Breach rate of PA-HMM, noMask and sensitiveMask on D11

Fig. 8 shows the distribution of published and suppressed data on D11 and D15. For PA-Markov, the bars show the fraction of sensitive or nonsensitive information in the whole dataset. For PA-HMM, since it can determine only whether to publish an event, the bars show

the fraction of events that contain sensitive information. We choose $\delta = 0.2$ for PA-HMM and $\delta = 0.65$ for PA-Markov and randomly tag sensitive information. PA-Markov and PA-HMM both publish some of the sensitive information or events with sensitive information. Based on Definition 4, publishing e_i preserves privacy if the prior probability of s exceeds $1 - \delta$ at position i. To explain that, if the adversary is very sure about some sensitive information, publishing or not would not shake his/her belief. However, it is counterintuitive for PA-Markov and PA-HMM to suppress a large number of nonsensitive information or events without sensitive information. According to Definition 4, although these events preserves privacy at its position, publishing them will breach δ-privacy at other positions. **Fig. 8** again demonstrates that PA-Markov and PA-HMM can protect against correlation attacks.

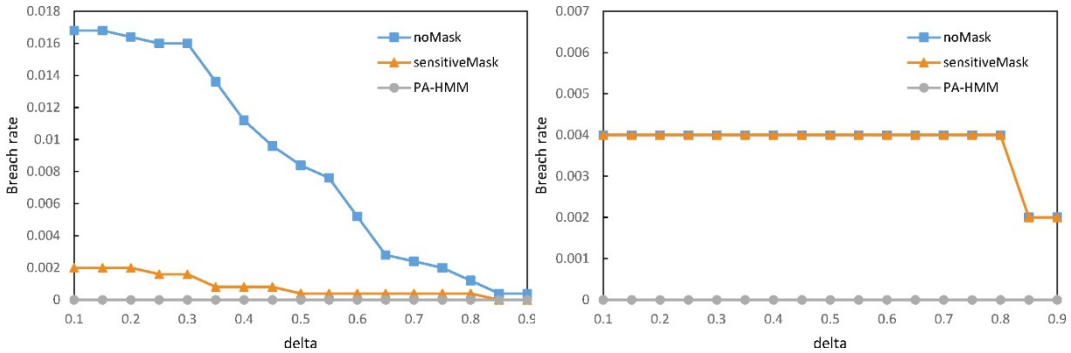
5.4.2 Utility

The utility of output datasets (for simplicity, the utility of an algorithm and the utility of its output are synonymous) by varying δ is shown in **Figs. 9** and **10**. For PA-Markov, we test the utility of four types of sensitive information. For PA-HMM, we test the utility when the number of latent states varies ($K = 4,6,8,10$). We observe that the variation trends of utility under different settings are similar. When we enhance privacy preservation, the utility decreases. This result implies that we must sacrifice some utility to satisfy privacy requirements in practical use. One proper tradeoff is setting a different δ for different users. We can assign a low δ for those who care more about their privacy (e.g., famous singers, actors) and a high δ for those who care less about their privacy (e.g., civilians). Contract theory seems to be a useful approach to balancing privacy and data utility [16].



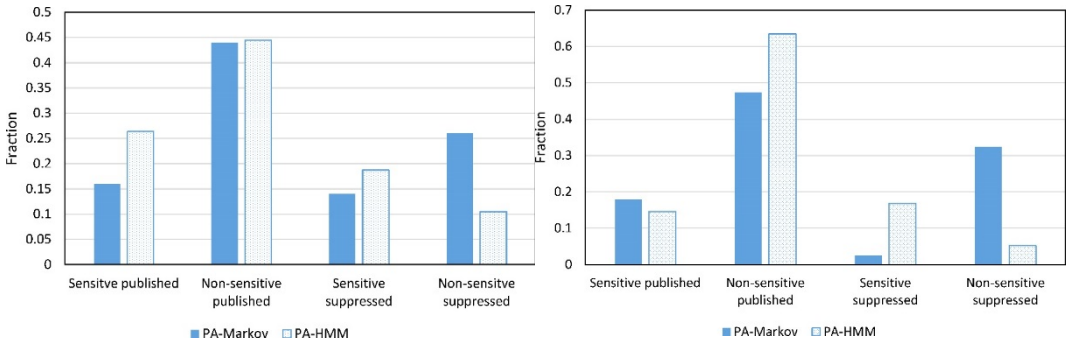
(a) Random a, t and l (b) Random a or t or l

Fig. 6. Breach rate of PA-Markov, noMask and nointernal on D15



(a) Random a, t and l (b) Random a or t or l

Fig. 7. Breach rate of PA-HMM, noMask and sensitiveMask on D15

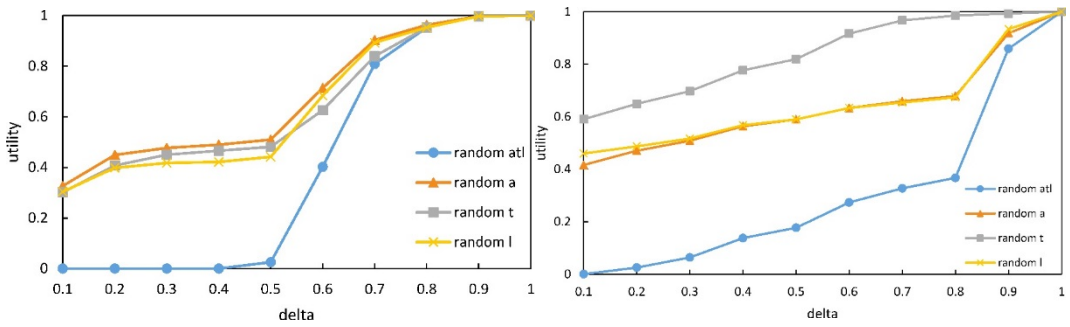


(a) D11 (b) D15

Fig. 8. The composition of published and suppressed data

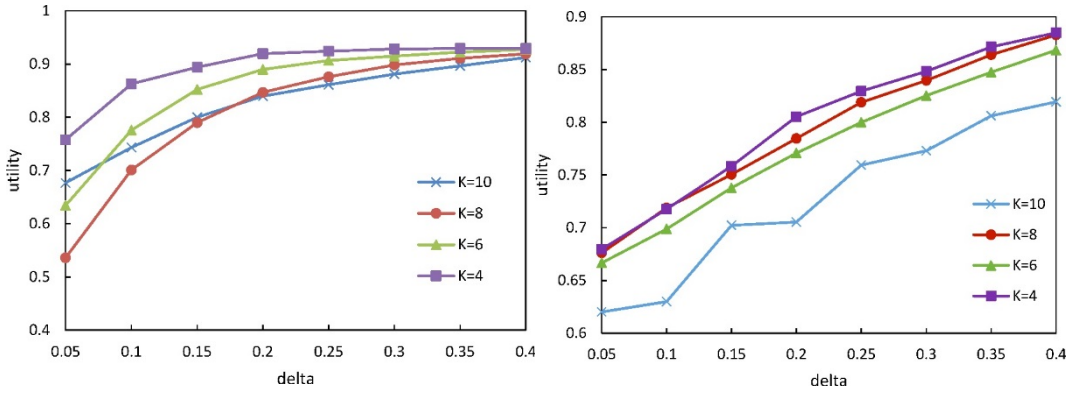
5.4.3 Running time

To compare the efficiency of PA-Markov, PA-HMM and the baselines, each algorithm is run for 10 times. Their average running time are shown in **Table 5**. PA-Markov and PA-HMM need much more time than sensitiveMask and noMask as they need multiple iterations while sensitiveMask and noMask take constant time. Additionally, after we conduct vectorSearching offline, the running time of PA-HMM is reduced significantly.



(a) D11 (b) D15

Fig. 9. Privacy-utility tradeoff of PA-Markov



(a) D11 (b) D15

Fig. 10. Privacy-utility tradeoff of PA-HMM

Table 5. Comparison of running time

	noMask	maskSensitive	PA-Markov	PA-HMM	PA-HMM (online)
D11	<1 ms	<1 ms	1 m	3 m	<1 ms
D15	<1 ms	<1 ms	58 s	3 m	<1 ms

6. Related Work

Recent years saw many inspiring works on PPDP. We provide a brief introduction of these works and compare S-PPATP with their approaches from the aspects of user model, adversary model, privacy requirement and data quality.

PPDP solutions can be broadly classified into two categories based on their attack principles [5]. The first category defends against linkage attacks, which considers that a privacy threat occurs when an attacker is able to link a record owner to a record in a published data set or to a sensitive attribute [17]. k-anonymity is a famous model to prevent linkage attack, which requires that each record is indistinguishable from at least k-1 other records. Li et al. [18] applied the partitioning-based and the clustering-based algorithms in the real-world privacy soccer fitness data publication to achieve the k-anonymity model. Gao et al. [19] propose a personalized k-anonymity model which take trajectory similarity and direction into account in the selection of anonymity set. Gurung et al. [20] and Dong et al. [21] adopted clustering-based anonymization algorithm to group similar trajectories and select representative trajectories. Their method can guarantee strict k-anonymity of published data, but probably reduced the utility of published data. Originally proposed for relational data, k-anonymity does not make assumption on the patterns of victims' data. Compared with these approaches, S-PPATP assumes that trajectory data indicate human mobility patterns which can be characterized by user model.

Many works have proposed improved technique for k-anonymity. It is found that in some cases, another type of privacy leakage, homogeneity attack, may still exist in k-anonymity data [17]. To address homogeneity attack, l-diversity has been proposed, which requires each sensitive attribute has to possess at least l distinct values in each anonymity group [9]. Wang et al. [22] proposed a novel privacy-preserving framework for LBS data publication. The framework considers the topological properties of the road network when providing privacy-preserving mechanisms for a single user and a batch of users. They also proposed two cloaking algorithms to achieve both k-anonymity and l-diversity. Zhu et al [23] proposed a noise technique to publish anonymized data and fulfilled the l-diversity requirements. Li. et al [24] proposed a data partitioning method in PPDP under the constraint of k-anonymity and l-diversity. However, l-diversity does not prevent attribute linkage attacks when the overall distribution of a sensitive information is skewed [5][25]. As a result, some works used a more strict privacy model called t-closeness to anonymize published data [26], which requires the distribution of sensitive information in any anonymity group to be close to the distribution in the overall dataset. To sum up, k-anonymity and corresponding improvement preserve privacy well only when the adversary has limited background knowledge about trajectory generator and provide undifferentiated protection for sensitive information. By contrast, S-PPATP solves a more serious problem that the adversary may have a background knowledge of user model and provides a flexible protection level by using δ-privacy.

The other category defends against probabilistic attack, which studies how the adversary changes the probabilistic belief on the privacy of a victim after accessing the published dataset [9]. Privacy preserving solutions of this category normally have a strong assumption on the

adversary's background knowledge. Gotz et al. [7] originally assumed that the adversary knows both the temporal correlations and the publishing algorithm. They proposed a framework, MASTIT, to filter user data that preserves δ-privacy. Li et al. [27] proposed a data publishing algorithm to prevent the attack based on a naive Markov user model. Gramaglia et al. [28] used both spatiotemporal generalization and suppression to ensure that the adversary gains little samples of the target user's trajectory with a significant data loss. S-PPATP belongs to this category. Compared with these works, S-PPATP extends the boundary of sensitive information and strengthens the adversary's power that he/she is aware of a victim's hidden life style (topic) and he/she can infer sensitive information by internal dependence inside an event. Additionally, S-PPATP applies suppression without generalization to ensure that the published data make more sense.

ϵ-Differential privacy is an extremely strict privacy model which was first proposed in [29]. Instead of comparing the prior probability and the posterior probability, ϵ-differential privacy proposes a strict requirement that the addition or removal of any single database record does not significantly influence the outcome of any inference. Some works applied ϵ-Differential privacy model to trajectory data publication [3][30][31]. ϵ-differential privacy seems to be an ultimate solution because it is proven that ϵ-differential privacy can protect against attackers with arbitrary background knowledge [29]. However, the privacy requirement is too rigorous for the LBS scenario [32]. Therefore, in S-PPATP, we relax the privacy requirement by ensuring that adversary's belief in sensitive information does not increase too much without taking further data removal or addition into account.

7. Conclusion and Future Work

In this paper, we propose a solution for PPATP, S-PPATP, which consists of modeling, algorithm design and algorithm adjustment. Although S-PPATP is an effective approach to privacy-preserving activity trajectories publishing, it can be further improved on each step. During modeling, more sophisticated user models can be used to better describe user behavior patterns, and a more powerful adversary can be assumed to have more background knowledge. During algorithm design, since we have discussed that neither PA-Markov nor PA-HMM is globally optimal in utility, a hybrid of PA-Markov and PA-HMM may further enhance the utility. During algorithm adjustment, the privacy-utility tradeoff is an important issue for data publishers and can be further studied for more different application scenarios.

References

[1] H. Ghasemzadeh, P. Panuccio, S. Trovato, G. Fortino, and R. Jafari, "Power-aware activity monitoring using distributed wearable sensors," *Human-Machine Systems, IEEE Transactions on*, vol. 44, no. 4, pp. 537-544, 2014. Article (CrossRef Link)

[2] B. Zhou, Q. Li, Q. Mao, W. Tu, and X. Zhang, "Activity sequence-based indoor pedestrian localization using smartphones," *Human-Machine Sys- tems, IEEE Transactions on*, vol. 45, no. 5, pp. 562-574, Oct 2015. Article (CrossRef Link)

[3] R. Chen, B. Fung, B. C. Desai, and N. M. Sossou, "Differentially private transit data publication: a case study on the montreal transportation system," in *Proc. of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM*, pp. 213-221, 2012. Article (CrossRef Link)

[4] J. Wan, C. Byrne, M. OGrady, and G. OHare, "Managing wandering risk in people with dementia," *Human-Machine Systems, IEEE Transactions on*, vol. 45, no. 6, pp. 819-823, Dec 2015. Article (CrossRef Link)

[5] B. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys (CSUR)*, vol. 42, no. 4, p. 14, 2010. Article (CrossRef Link)
[6] B. Ağır, K. Huguenin, U. Hengartner, and J.-P. Hubaux, *On the privacy implications of location semantics*, Ph.D. dissertation, EPFL, 2015.
[7] M. Go¨tz, S. Nath, and J. Gehrke, "Maskit: privately releasing user context streams for personalized mobile applications," in *Proc. of the 2012 ACM SIGMOD International Conference on Management of Data. ACM*, pp. 289-300, 2012. Article (CrossRef Link)
[8] K. Zheng, S. Shang, N. J. Yuan, and Y. Yang, "Towards efficient search for activity trajectories," in *Proc. of Data Engineering (ICDE), 2013 IEEE 29th International Conference on. IEEE*, pp. 230-241, 2013. Article (CrossRef Link)
[9] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 3, 2007. Article (CrossRef Link)
[10] C. Song, Z. Qu, N. Blumm, and A. Baraba´si, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018-1021, 2010. Article (CrossRef Link)
[11] A. Mannini and A. M. Sabatini, "Accelerometry-based classification of human activities using markov modeling," *Computational intelligence and neuroscience*, vol. 2011, p. 10, 2011. Article (CrossRef Link)
[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.
[13] A. Gruber, Y. Weiss, and M. Rosen-Zvi, "Hidden topic markov models," in *Proc. of International Conference on Artificial Intelligence and Statistics*, pp. 163-170, 2007.
[14] M. Go¨tz, *On user privacy in personalized mobile services*, Ph.D. dissertation, Cornell University, 2012.
[15] A. Arasu, M. Go¨tz, and R. Kaushik, "On active learning of record matching packages," in *Proc. of the 2010 ACM SIGMOD International Conference on Management of data. ACM*, pp. 783-794, 2010. Article (CrossRef Link)
[16] L. Xu, C. Jiang, Y. Chen, Y. Ren, K. R. Liu, "Privacy or utility in data collection? a contract theoretic approach," *IEEE Journal of Selected Topics in Signal Processing*, 9 (7), 1256-1269, 2015. Article (CrossRef Link)
[17] S. Yu, "Big privacy: Challenges and opportunities of privacy study in the age of big data," *IEEE access*, 4, 2751-2763, 2016. Article (CrossRef Link)
[18] R. Li, S. An, D. Li, J. Dong, W. Bai, H. Li, Z. Zhang, Q. Lin, "K-anonymity model for privacy-preserving soccer fitness data publishing," in *Prco. of MATEC 565 Web of Conferences*, Vol. 189, p. 03007, 2018. Article (CrossRef Link)
[19] S. Gao, J. Ma, C. Sun, and X. Li, "Balancing trajectory privacy and data utility using a personalized anonymization model," *Journal of Network and Computer Applications*, vol. 38, pp. 125–134, 2014. Article (CrossRef Link)
[20] S. Gurung, D. Lin, W. Jiang, A. Hurson, R. Zhang, "Traffic information publication with privacy preservation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5 (3), 44, 2014. Article (CrossRef Link)
[21] Y. Dong, D. Pi, "Novel privacy-preserving algorithm based on frequent path for trajectory data publishing," *Knowledge-Based Systems*, 148, 55-65, 2018. Article (CrossRef Link)
[22] Y. Wang, Y. Xia, J. Hou, S.-m. Gao, X. Nie, Q. Wang, "A fast privacy-preserving framework for continuous location-based queries in road networks," *Journal of Network and Computer Applications*, 53, 57-73, 2015. Article (CrossRef Link)
[23] H. Zhu, S. Tian, M. Xie, M. Yang, "Preserving privacy for sensitive values of individuals in data publishing based on a new additive noise approach," in *Proc. of 2014 23rd International Conference on Computer Communication and Networks (ICCCN), IEEE*, pp. 1-6, 2014. Article (CrossRef Link)

[24] S. Li, H. Shen, Y. Sang, H. Tian, "An efficient method for privacy-preserving trajectory data publishing based on data partitioning," *The Journal of Supercomputing*, 1-25, 2019. Article (CrossRef Link)
[25] P. R. M. Rao, S. M. Krishna, A. S. Kumar, "Privacy preservation techniques in big data analytics: a survey," *Journal of Big Data*, 5(1), 33, 2018. Article (CrossRef Link)
[26] Z. Tu, K. Zhao, F. Xu, Y. Li, L. Su, D. Jin, "Protecting trajectory from semantic attack considering k-anonymity, l-diversity, and t-closeness," *IEEE Transactions on Network and Service Management*, 16(1), 264-278, 2018. Article (CrossRef Link)
[27] X. Li, S. Wei, and G. Sun, "A scheme for activity trajectory dataset publishing with privacy preserved," in *Proc. of UIC-ATC-ScalCom-CBDCom-IoP 2015. IEEE*, pp. 247-254, 2015. Article (CrossRef Link)
[28] M. Gramaglia, M. Fiore, A. Tarable, A. Banchs, "Preserving mobile subscriber privacy in open datasets of spatiotemporal trajectories," in *Proc. of IEEE INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pp. 1-9, 2017. Article (CrossRef Link)
[29] D. Cynthia, "Differential privacy," *Automata, Languages and Programming*, 1-12, 2006. Article (CrossRef Link)
[30] Y. Xiao, L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proc. of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM*, pp. 1298-1309, 2015. Article (CrossRef Link)
[31] Q. Miao, W. Jing, H. Song, "Differential privacy-based location privacy enhancing in edge computing," *Concurrency and Computation: Practice and Experience*, 31(8), e4735, 2019. Article (CrossRef Link)
[32] K. Chatzikokolakis, C. Palamidessi, M. Stronati, "A predictive differentially-private mechanism for mobility traces," in *Proc. of International Symposium on Privacy Enhancing Technologies Symposium, Springer*, pp. 21-41, 2014. Article (CrossRef Link)

Xianming Li received the bachelor degree in computer science from University of Science and Technology of China (USTC) in 2011. He has been a Ph.D. student in National High Performance Computing Center (Hefei) since 2011. Currently he is a member of Algorithm and Data Application (Ada) Research Group. His research interests include trajectory data mining, human behavior, game theory and privacy-preserving data publishing.

Guangzhong Sun obtained Ph.D. in computer science of University of Science and Technology of China (USTC) in 2005. He has been a visiting scholar at Yale University (2014 to 2015), Microsoft Research Asia (2010 to 2011 and 2007 to 2008), and Intel China Research Center (Jul. to Dec., 2006). Currently he is an associate professor in School of Computer Science and Technology, USTC. He is also a member of National High Performance Computing Center (Hefei) and the head of Algorithm and Data Application (Ada) Research Group.