文章编号：2095-2163(2019)02-0009-07 中图分类号：TP311.13 文献标志码：A

基于噪音前缀树的轨迹数据发布隐私保护算法研究

石秀金，徐嘉敏，王 锐，张姝俪
（东华大学 计算机科学与技术学院，上海 201620）

摘 要：现有的轨迹数据发布隐私保护算法存在轨迹距离短、部分轨迹序列丢失的弊端，因此提出一种基于噪音前缀树的轨迹数据发布隐私保护算法。该算法首先从轨迹离散化处理得到的位置中提取特征点并为其添加一定量的拉普拉斯噪音，然后用噪音干扰后的特征点校准原始轨迹得到校准轨迹并以此构建前缀树，通过向节点添加拉普拉斯噪音形成最终满足隐私需求的发布版本。相较于已有算法，提出一种创新的轨迹数据发布算法，并且考虑了发布数据的隐私性和实用性。通过实验验证了所提算法的有效性和相较于传统算法的优越性。

关键词：轨迹数据；大数据发布；噪音前缀树；隐私保护；拉普拉斯噪音

Research on privacy protection algorithm of trajectory data distribution based on noise prefix tree

SHI Xiujin, XU Jiamin, WANG Rui, ZHANG Shuli
(School of Computer Science and Technology, Donghua University, Shanghai 201620, China)

【Abstract】 The existing privacy protection algorithms of trajectory data release have the drawbacks that the distance of the protected trajectory is shorter than origin trajectory and part of the trajectory sequence is lost. So it proposes a privacy preserving algorithm based on noise prefix tree for trajectory data dissemination. Firstly, extract the feature points from the positions obtained by trajectory discretization and add Laplace noise to it; then calibrate the original trajectory with noise-disturbed feature points to get the calibration trajectory and build a noise prefix tree with the calibration trajectory. Laplace noise is added to the leaf nodes to ensure privacy and security. Compared with existing algorithms, the paper proposes an innovative trajectory data distribution algorithm, and considers the privacy and practicality of published data. Experiment shows that the algorithm is effective and performs better than existing algorithms.

【Key words】 trajectory data; data release; noise prefix tree; privacy protection; Laplace noise

0 引 言

随着定位技术的快速发展和广泛应用，越来越多的轨迹数据被产生、发布、采集和分析。如车载定位系统，带 GPS 的移动设备以及位置传感器等，这些设备在提高人们生活质量的同时也产生了大量的轨迹数据，其中可能包含车辆、身份以及实时位置等诸多数据，然而这些内容都含有个人敏感信息，直接发布会给个人隐私造成威胁。因此，对轨迹数据发布的隐私保护具有重要的研究意义。

在轨迹数据的发布中，最简单的方式是删除轨迹上的准标识符属性[1]，如个人基本信息，但却不能全面保护移动对象的轨迹隐私。k-匿名模型[2-3]能够在一定程度上保护轨迹隐私，但是该模型具有容易受到新型攻击、系统开销大等缺陷，因此随着数据的大量收集，这种模型并不能有效保证轨迹信息。

文献[4]中证明采用差分隐私保护模型，使用合理数量的噪音保护位置数据中的敏感信息，而且还能保证数据的可用性。差分隐私方法提供了严格的隐私模型来保护空间信息中的敏感数据，同时能够保证发布数据的实用性，也并不关心攻击者拥有的背景知识，该模型向查询或者分析结果中添加一定量的噪音来达到隐私保护的效果。差分隐私模型可以应用于数据发布、数据挖掘、个性化推荐等诸多领域。

对于轨迹数据的发布，已有不少研究者考虑到用差分隐私模型来实现轨迹数据发布的隐私保护。2012 年，Chen 等人[5]提出一种基于噪音前缀树的交通数据发布隐私保护方式，并通过实验证明这种方式可以应用于轨迹数据发布的隐私保护研究领域。文献[6]提出基于差分隐私保护的轨迹序列非交互式合成方式，然而这种方式局限于短距离内的

作者简介：石秀金(1975-)，男，博士，副教授，主要研究方向：大数据、隐私保护、移动互联网应用等；徐嘉敏(1993-)，女，硕士研究生，主要研究方向：差分隐私保护。

收稿日期：2018-11-30

轨迹数据隐私保护,在更大规模的空间区域和实际情况中是无效的。2012 年,Chen 等人[7]又提出通过 n-gram 思想解决短距离问题,然而这种方式导致了轨迹序列的丢失。

在综合上文研究成果后,本文提出了一种基于噪音前缀树的轨迹数据隐私保护算法。该算法分 3 步,可描述为:首先将轨迹数据离散化后通过聚类分析获得具有某些特定性质的特征点(本文称为锚点),对锚点进行隐私保护后构建参考系;然后基于该参考系对原始轨迹进行校准,并用校准轨迹构造噪音前缀树;最后根据拉普拉斯机制向叶子节点中添加噪音,形成用于发布的数据版本。该算法在一定程度上解决了空间限制以及序列丢失等问题。

1 问题描述

轨迹数据是一组从实际路线中提取的位置有界和时间有序的序列,可以表示为: $t_j = \{(x_1,y_1,t_1),(x_2,y_2,t_2),\cdots,(x_n,y_n,t_n)\}$。其中,$(x_i,y_i,t_i)$ $(1 \leqslant i \leqslant n)$ 表示移动对象在 t_i 时刻的位置为 (x_i,y_i),也称为采样位置或采样点,t_i 则称为采样时间。若忽略具体采集到某个位置点的时间,而是在指定的时间段内按照时间序列进行排序,同时以 p_1 表示 (x_1,y_1) 代表的位置点,则轨迹 t_j 可以表示为 $t_j = \{p_1,p_2,\cdots,p_n\}$。其中 $p_i \in P$,P 表示一定区域内站点位置的集合,$|P|$ 表示站点位置的数量且 $i \leqslant |P|$。

例如,假设 $p_1(x_1,y_1)$ 表示某小区,$p_2(x_2,y_2)$ 表示某地铁站,$p_3(x_3,y_3)$ 表示某商场,$p_4(x_4,y_4)$ 表示某学校,$p_5(x_5,y_5)$ 表示某条路的第二个交叉路口,那么轨迹 $t_j = \{(x_1,y_1,t_1),(x_2,y_2,t_2),(x_3,y_3,t_3),(x_1,y_1,t_4)\}$ 表示:$t_1 \sim t_4$ 时间段内,车辆 A 从小区出发,先到达地铁站,然后到达商场,最后在 t_6 时刻回到小区。也可以表示为:在 $t_1 \sim t_4$ 时间段内,$t_j = \{p_1,p_2,p_3,p_1\}$。

在指定时间内,如以一周为期限,每次采样的时长为 2 h,每周分别在周一和周三上午 10:00~12:00 进行采样,该车辆可能有不止一条轨迹数据产生,用 $IT_q = [t_{j1},t_{j2},\cdots,t_{jk}]$ 表示该车在一周内产生的轨迹数据。该周内共产生的轨迹数据集为 $T = [IT_1,IT_2,\cdots,IT_m]$。研究得到的由全部轨迹组成的一般的轨迹数据集可见表 1。

考虑到后续的研究需要,这里关于文中涉及到的基础概念理论将展开探讨论述如下。

定义 1 锚点 空间域中相对稳定的特征位置点,其变化不大。如基于道路分析提取到的交叉点、转折点,基于轨迹数据中的位置进行聚类分析提取到的质心等。锚点用 a_i 表示,数量上小于该数据集的位置总数 $|P|$,锚点组成的集合称为锚点集 A,用来构建参考系。

表 1 轨迹数据集

Tab. 1 Trajectory data set

序号	用户	轨迹号	轨迹路径
1	车辆 A	t_{j_1}	$p_1 \rightarrow p_2 \rightarrow p_3$
2	车辆 B	t_{j_1}	$p_1 \rightarrow p_2$
3	车辆 C	t_{j_1}	$p_3 \rightarrow p_2 \rightarrow p_1$
4	车辆 A	t_{j_2}	$p_1 \rightarrow p_2 \rightarrow p_4$
5	车辆 B	t_{j_2}	$p_1 \rightarrow p_2 \rightarrow p_3$
6	车辆 B	t_{j_3}	$p_3 \rightarrow p_2$
7	车辆 A	t_{j_3}	$p_3 \rightarrow p_2$
8	车辆 C	t_{j_2}	$p_3 \rightarrow p_1$

定义 2 轨迹校准 基于由集合 A 构成的参考系,校准过程就是将原始轨迹 $t_j = [p_1,p_2,\cdots,p_n]$ 转换为 $t'_j = [a_1,a_2,\cdots,a_m]$。其中,$a_i \in A$,$1 \leqslant i \leqslant m$,$m$ 可以等于 n。称 t'_j 是 t_j 的校准轨迹。

定义 3 差分隐私[4,8-10] 给定数据集 D 和 D',且 D 和 D' 之间最多相差一条记录,即 $|D\Delta D'| \leqslant 1$。给定一个隐私算法 A,$Range(A)$ 表示 A 的取值范围,若算法 A 在数据集 D 和 D' 上任意输出结果 $O(O \in Range(A))$ 满足下列不等式,则 A 满足如下的 ε - 差分隐私:

$$Pr[A(D) = O] \leqslant e^{\varepsilon} * Pr[A(D') = O], \quad (1)$$

其中,概率 $Pr[\cdot]$ 表示隐私的披露风险,由算法 A 的随机性控制;隐私预算参数 ε 表示算法 A 所能提供的隐私保护程度。ε 越大,隐私保护程度越低,反之,隐私保护程度越高。

由式(1)可以看出,差分隐私保护模型是基于 2 个只有一条记录的不同数据集进行数据失真处理的,结果是保证即使攻击者知道大量的背景知识也无法准确判断该记录是否在表中。

研究可知,差分隐私的组合特性[11]可分为 2 种,对其可做阐释解析如下。

(1)顺序组合。给定数据集 D 以及一组关于 D 的差分隐私 $A_1(D),A_2(D),\cdots,A_m(D)$,分别满足 ε^- 差分隐私,且任意 2 个算法的随机过程相互独立,则这些算法组合起来的算法满足 $\sum_{i=1}^{m} \varepsilon^-$ 差分隐私。

(2)并行组合。$A_1(D),A_2(D),\cdots,A_m(D)$,分别表示输入集为 $D_1,D_2,\cdots,D_m$ 的一系列满足 ε^- 差分隐私算法,且任意 2 个算法的随机过程相互独立。

则这些算法组合起来的算法满足 ε^- 差分隐私。

定义 4　噪音前缀树[12]　序列数据集 T 的前缀树 PT 是三元组 $PT=(V,E,Root)$，其中 V 是标记位置的节点集合，每个节点对应于 T 中唯一的前缀；E 是边的集合，表示节点之间的转换；$Root \in V$ 是 PT 的虚根。

定理 1　拉普拉斯机制[12]　拉普拉斯噪音实质上是一组满足拉普拉斯分布的随机值，其原理是向原始数据及统计分析结果中添加服从拉普拉斯 $lap(b)$ 的噪音，实现加入噪音后的查询结果满足差分隐私约束效果。文献[8]提出拉普拉斯机制，则需要输入数据集 D，函数 f 和隐私参数 α。所添加的噪声符合概率密度函数 $p(x\mid\lambda)=\frac{1}{2\lambda}e^{-\frac{|x|}{\lambda}}$ 的拉普拉斯分布，其中 λ 由全局灵敏度 f 和预期隐私参数 α 确定。

定义 5　地理相似性[13]　给定 2 条轨迹 $t_1=[p_{11},p_{12},\cdots\cdots,p_{1n}]$ 和 $t_2=p_{21},p_{22},\cdots\cdots,p_{2m}$，地理相似性表示为：

$$geoDist(t_1,t_2)=Dist(ctr(t_1),ctr(t_2))+Dist(ctr(t_1),ctr(t_2))*\frac{||t_1|-|t_2||}{\max(|t_1|,|t_2|)}-\frac{|DPL_1|-|DPL_2|}{2}*\cos(DPL_1,DPL_2),\quad(2)$$

其中，$ctr(t_1)=\left(\frac{\sum_{i=1}^{n-1}(x_{i+1}^2-x_i^2)}{2*\sum_{i=1}^{n-1}(x_{i+1}-x_i)},\frac{\sum_{i=1}^{n-1}(x_{i+1}^2-y_i^2)}{2*\sum_{i=1}^{n-1}(y_{i+1}-y_i)}\right)$，$DPL_1$ 是从起始位置到终点位置的短切向量，$\cos(DPL_1,DPL_2)$ 是 DPL_1 和 DPL_2 角度的余弦。

定义 6　召回率　基于点场景的召回率用于衡量位置点的隐私保护级别，满足如下公式：

$$Recall_{pot}=\frac{Count\ of\ IST}{Count\ of\ anchor\ points\ in\ the\ CT},\quad(3)$$

其中，CT 是基于 NDBSCAN 算法得到的扰动锚点的集合；IST 是扰动点与原始位置点之间的距离小于初始阈值的点。召回率越大，则认为数据的可用性越大。

基于轨迹场景的召回率用于衡量对轨迹数据的隐私保护级别，满足如下公式：

$$Recall_{tra}=\frac{Count\ of\ IST}{Count\ of\ trajectories\ in\ the\ CT},\quad(4)$$

其中，IST 是被重新识别的轨迹，CT 是基于 Final Private Trajectory Release 算法得到的发布数据集合。召回率越高，则认为数据的实用性越大。

定义 7　全局敏感性[4]　对于任意一个函数 $f:D\rightarrow R^d$，函数 f 的全局敏感性为：

$$\Delta f=\max\|f(D)-f(D')\|_p.\quad(5)$$

其中，D 和 D' 之间最多相差一条记录；R 表示所映射的实数空间；d 表示函数 f 的查询维度；p 表示度量 Δf 使用的 L_p 距离，通常用 L_1 来度量。

2　算法及性能分析

传统基于差分隐私的轨迹数据发布算法中，基于可变长度 n-gram 的 SD 算法通过噪声合成的方法改进轨迹数据发布的隐私保护效率，然而 n-gram 模型只是简单地将所有序列切割成至少 $I_{\max}$ 长度的序列构建树，对长距离轨迹会丢失一定的信息。本文的主要目的旨在提出一种基于差分隐私保护机制的轨迹数据发布算法以解决上述局限性。

该算法可分为 2 个阶段，对此则分述如下。

(1)在第一阶段，为了从大量的轨迹位置点中提取具有一定特征的锚点构建参考系，本文在 DBSCAN 算法[14]的基础上进行改进和丰富，提出 NDBSCAN 算法。NDBSCAN 算法将全部数据点分为核心点(半径 r 内含有超过阈值 τ 的点)和非核心点。通过邻域查询不断寻找当前点的种子点，并用新种子扩展同一类别的集群，直到全部的种子点用完，得到全部的特征点，再为其添加拉普拉斯噪音。

(2)在第二阶段，主要任务是使用上述扰动锚点对原始轨迹进行校准，得到校准轨迹。而后基于差分隐私框架构造噪音前缀树，在树的叶子节点添加拉普拉斯噪音形成最终的发布版本。然而噪音的大量添加使得发布的数据实用性降低，为了添加更少的噪音，该算法通过滤波器来减少空节点的数量。算法的主要思想是：从虚拟根开始，逐级分析每一层高度上的节点，通过估计当前节点的子树高度来分配隐私预算并计算滤波器在该节点的截止值。对于添加噪音后能达到截止值的非空节点，将其添加到前缀树中生成最终发布版本。

这 2 个阶段都需要添加拉普拉斯噪音。第一阶段通过基于先验知识的阈值进行控制；第二阶段提出一种自适应算法解决构建前缀树过程中的噪音计数问题。

2.1　算法基本思想

本文算法首先是基于道路分析提取轨迹数据中

的交叉点,转折点,对轨迹数据离散化得到的位置点进行基于密度的聚类算法分析(NDBSCAN)提取轨迹关键点,由交叉点、转折点和轨迹关键点组成特征点集,本文称为锚点集。出于隐私性考虑,本文将对锚点集中的特征点进行隐私保护操作,为其添加一定量的拉普拉斯噪音。在位置点获得隐私保证的基础上,基于扰动锚点校准原始轨迹。通过对原始轨迹进行校准,去除冗余位置点,在不影响原始轨迹大致走向的前提下缩短了长度,得到校准轨迹。基于校准轨迹构建前缀树,从而降低了前缀树构造过程中的计算量和算法复杂度。

然后,基于校准轨迹构造噪音前缀树,该树从根节点到每一个叶子节点的路径表示一条轨迹数据。本文基于一种自适应剪枝方案处理构建前缀树的过程中产生的空节点,以提高效率和效用。对于空节点,本文采取的一种解决方案是另外存储,算法结束时再做统一处理,对于非空节点,为其添加拉普拉斯噪声并计算噪音计数,对于大于阈值的节点,将其添加到树中,否则将其标记为叶子节点。通过自适应剪枝方案减少了前缀树构建过程中的计算量,提升了算法性能。

2.2 算法描述

轨迹数据发布是从轨迹位置点中提取锚点,而后基于该锚点校准原始轨迹后构建噪音前缀树生成发布版本。用于提取锚点的 NDBSCAN 算法和构建噪音前缀树的 FPTR 算法分别详见如下代码设计。

算法 1 NDBSCAN 算法

输入:轨迹 t_j 上全部位置点的集合 P;存储每个集群的结果 RL;存储种子点的队列 SD;邻域查询 RQ 的半径 r;核心点的阈值 τ;存储全部锚点的集合 $M=\{\}$;M 中的一组位置点集 M_{cj};锚点的计数 CT;锚点的噪音计数 CT'

输出:扰动锚点集

Construct R-Tree index over P
FOR each point p in P, do
  IF $P[i].clustered = false$ THEN
  $SD.add(P)$;
  $RL.add(P)$;
    while SD is not null do
    $Points\ P' = SD.pop()$;
    $Points\ P' = RQ(P', r)$;
    FOR $i = 0$ to $|P'|$ do
    IF $P'[i].clustered = false$ and collect with P' directly
      $RL.add(P'[i])$;
      $SD.add(P'[i])$;
    end
  end
  end
  IF $|RL| \geqslant \tau$ THEN
    $M.add(RL)$;
    FOR each point P'' in RL do
    $P''.clustered = true$
    end
  end
 $RL.clear()$
 end
 FOR $j = 1$ to $|M|$ do
 $CT' = |M_{cj}| + lap(\sigma_{ct})$;
 $IF\ CT' > \gamma$ then

$$CC_i = \frac{\sum_{k=1}^{|M_{cj}|}(x_k, y_k)}{|M_{cj}|}$$

 $CC' = NoisyLap(\sigma^j)(CC_j)$
End

分析可知,NDBSCAN 算法是从轨迹 t_j 的全部位置点集合的某一点 P 开始,若判断 P 为核心点,则将与 P 同类别的点即都归并作为 P 的邻域点,而且将这些点作为种子点进行下一轮考察,不断扩展种子点所在的类直至找到完整类。重复以上步骤直至寻找到其它类,则算法结束。对算法获取的锚点集添加拉普拉斯噪音形成扰动锚点集。

算法 2 **Final Private Trajectory Release 算法**

输入:校准后的轨迹数据集 CT, 全部的隐私预算 ε, 初始阈值 θ, 轨迹的最大长度 $x_{\max}$

输出:噪音前缀树 PT

Initialize a prefix tree PT with a virtual root
FOR $i = 0; i < x_{\max}; i++$ do
FOR $j = 0; j < |nds|; j++$ do
  IF $nds[j].flag$ is $false$
  $tr(nds[j]).height = EstimateHeight()$;
  $\varepsilon[i][j] = PBD(tr(nds[j]).height)$;
  $\theta[i][j] = Threshold(\varepsilon[i][j])$;
  $nodes[pcn] = All\ possible\ children\ of\ nds[j]$;
    FOR $k = 0$ to $|pcn|$ do
    $Count = c(pcn[k])$;
    IF $NoisyCount \geqslant \theta$ then
    $NoisyCount = c'(pcn[k])$

```
    NoisyCount ≥ θ then
    Add pcn[k] into PT;
   else
    pcn[k].flag = true;
    end
   else
Node[empty].push(pcn[n]);
Node[empty´] = ThresholdSampling(empty´);
   end
   end
   For m = 0 to | empty´ | do
    NoisyCount = c´(empty´[m]);
    IF NoisyCount ≥ θ then
        Add empty´[m] into PT;
   else empty´[m].flag = &;
        empty´[m].flag = &;
   end
   end
  end
End
```

分析可知，FinalPrivateTrajectoryRelease（FPT）算法是在文献[7]中提出的噪音树构建算法的基础上加以改进的，该算法能够自适应剪枝以减少噪音的添加。该算法的输入是校准轨迹数据集 CT、隐私预算 ε、初始阈值 θ 和轨迹的最大长度 $x_{\max}$。首先，初始化前缀树 PT 并创建虚拟根节点，构建噪音树是找到所有序列，直到计数大于阈值而长度小于 $x_{\max}$，迭代生成树中每个层级的节点并对每个节点进行判断：如果该节点不是叶子节点，则估计该节点子树的高度并基于自适应隐私预算分配算法计算该节点应被分配的 ε 值。此外，找到该节点的所有可能子节点，同时计算在数据集 CT 中从根节点到该节点轨迹前缀的频率。在此过程中，为了处理大量空节点造成的冗余现象，本文提出一种自适应剪枝方案处理空节点和非空节点的潜在子节点。对于计数非 0 的非空节点，通过拉普拉斯噪音添加到实数中来计算噪音计数，对于噪声计数大于阈值的节点添加到噪音前缀树中，否则该节点被标记为叶节点；对于计数为 0 的空节点，则将其专门存储起来，不会添加到噪音前缀树中，这样在一定程度上就减少了构造噪音前缀树的计算量。如果噪音前缀树的高度达到 $n_{\max}$ 或者没有可添加的节点时，遍历过程即停止，其噪音计数无法超过阈值，或者节点所在高度的层级隐私预算消耗完。

2.3　算法性能分析

这里有 2 个连续的步骤，分别是：隐私参考系统和隐私轨迹发布。本文首先证明这 2 个连续步骤生成的发布结果分别满足 ε^- 差分隐私，再通过差分隐私的组成属性证明该算法满足 ε^- 差分隐私。其中，ε_{pr} 用于隐私参考系统，并且 ε_{pg} 用于隐私轨迹发布。

对于第一阶段，又分为计数扰动 ε_{ct} 和质心扰动 ε_{cc}。在此，$\varepsilon_{pr} = \varepsilon_{ct} + \varepsilon_{cc}$，计数灵敏度，即 Δf^j_{ct} 为 $\max(NUM_{individual}(points))$，质心灵敏度即 Δf^j_{cc} 为 $\max(distance(p_i, p_j))/2$。因此，通过将 $Laplace\left(\frac{\Delta f^j_{ct}}{\varepsilon_{ct}}\right)$ 函数的随机噪声添加到每个聚类计数，将 $Laplace\left(\frac{\Delta f^j_{cc}}{\varepsilon_{cc}}\right)$ 函数的随机噪声添加到每个聚类质心使得算法 1 满足 $(\varepsilon_{ct} + \varepsilon_{cc})^-$ 差分隐私。

对于第二阶段，令 $l_{\max}$ 为原始轨迹数据集 T 中的最长序列，因此，增加或者去除一个单独轨迹 (T') 最多影响 $l_{\max}$ 条根到叶子节点的序列，记为 $SE(l_{\max})$，即 $\Delta f = l_{\max}$。定义噪音性强轨迹的功能函数 $M(T)$，那么在噪音前缀树 PT 中至少有 $|P|x$ 个节点，记为 X，分配给 $nd \in X$ 的隐私预算是 ε_{nd}。采用拉普拉斯噪声机制，可以表示为：

$$\frac{Pr(M(T) = PT)}{Pr(M(T') = PT)} = \prod_{nd \in X} \frac{\exp\left(-\varepsilon_{nd} \frac{|c'(nd) - c'(nd,T)|}{l_{\max}}\right)}{\exp\left(-\varepsilon_{nd} \frac{|c'(nd) - c'(nd,T')|}{l_{\max}}\right)} \leqslant$$

$$\exp\left(\sum_{nd \in X} \varepsilon_{nd} |c'(nd) - c'(nd,T')|\right) =$$

$$\exp\left(\frac{\sum_{i=1}^{l_{\max}} \sum_{nd \in SE(i)} \varepsilon_{nd}}{l_{\max}}\right) \leqslant$$

$$\left(\frac{\sum_{i=1}^{l_{\max}} \varepsilon_{pg}}{l_{\max}}\right) \leqslant \exp(\varepsilon_{pg})$$

其中，$|c'(nd) - c'(nd,T')|$ 是 (T') 变化后根到叶子节点的轨迹计数，$\sum_{i=1}^{l_{\max}} \varepsilon_{nd} \leqslant \varepsilon_{pg}$，$\varepsilon_{pg}$ 是轨迹数据发布的隐私预算。因此，对每个节点计数执行 $Lap\left(\frac{l_{\max}}{\varepsilon_{nd}}\right)$ 满足 ε^- 差分隐私。

3　实验分析

为了验证算法的隐私性和实用性，本文通过大

量实验来评估 FPT 算法的性能。本文的实验分为 2 部分。第一部分:验证锚点集和发布轨迹的隐私性;第二部分:验证发布轨迹数据的实用性。实验环境为:Inter(R)Core(TM)i5-2450M CPU@2.50 GHz,8 Gb 内存,Win7 操作系统,编程环境为 MyEclipse。实验数据集为 Gowalla 和 Brightkite 上的签到数据[11],测试数据集由 Thomas Brinkhoff 路网移动节点数据生成器生成。

3.1 隐私性验证

从清洗过的数据集中分别随机选择 8 000 个轨迹,其中位置点的总数是 252 448,最大轨迹长度是 192,平均轨迹长度为 13。设置不同的 $\varepsilon = 0.01$、0.1、0.5、1.0、1.5,设置聚类算法的初始阈值为半径 0.01 km内至少包含 50 个特征点。每组实验进行 10 次,并以平均值作为该组实验的最终结果。

实验中,设置欧氏距离阈值为 100 m,隐私预算 $\varepsilon = 0.01$、0.1、0.5、1.0、1.5 时,对 2 个数据集分别基于本文算法提取的锚点以及发布轨迹的召回率对比如图 1 所示。其中,横坐标表示隐私预算,纵坐标表示召回率。

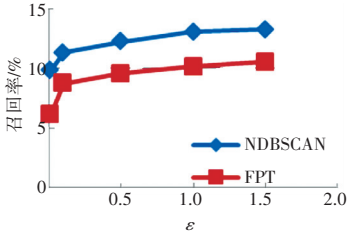


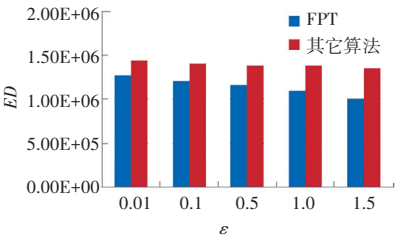
图 1 经过本文算法处理后数据的召回率

Fig. 1 Data recall rate of this algorithm

图 1 表示数据集基于本文算法发布的锚点以及数据在给定阈值的前提下所能达到的召回率。分析图 1 可知,随着 ε 变大,锚点的召回率和发布轨迹的召回率都呈现增加的趋势。这是由于 ε 越大,隐私保护程度越低,添加的拉普拉斯噪音越少,锚点和轨迹的可识别率越高,因此召回率随之增加。反之,ε 越小,隐私保护程度越高,可识别的锚点和轨迹越少,相应的召回率越低。从实验结果来看,即使在弱隐私保护级别下,攻击者无法重新识别的敏感位置百分比超过 86 %,无法重新识别的轨迹百分比超过 89%,证明了本文算法的隐私性。

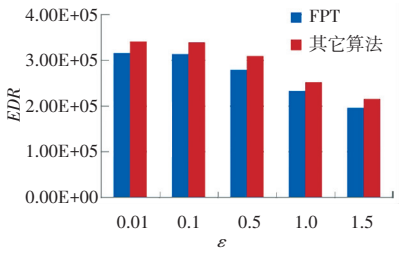
3.2 实用性验证

为了评估数据的实用性,本文基于 2 个数据集针对隐私保护后发布的数据进行 KNN 查询以及频繁序列模式挖掘,并与文献[7] 中提出的直接构建噪音前缀树实现轨迹数据隐私保护的算法进行对比分析。本文采用欧式距离(*ED*) 以及实序列编辑距离(*EDR*) 进行计算,欧氏距离用于评估具有相同长度轨迹相似性,实序列编辑距离的目的则是匹配每个可能的位置对 (p_i, p_j) 来计算使得 t_1 和 t_2 等效所需的最小编辑数。其中,$p_i \in$ 原始轨迹数据集,$p_j \in$ 校验轨迹数据集。当 p_i, p_j 匹配时,编辑距离为 0,否则为 1。从数据集中随机选择 500 个随机轨迹执行 KNN 查询实验对比如图 2(a)、(b)所示。其中,横坐标表示隐私预算,纵坐标表示频繁模式挖掘的距离。



(a)进行 KNN 查询的欧式距离 *ED*

(a) Euclidean distance *ED* for KNN query



(b)进行 KNN 查询的编辑距离 *EDR*

(b) Edit distance *EDR* for KNN query

图 2 不同隐私保护水平下的相对误差

Fig. 2 Relative error at different levels of privacy protection

从实验结果可以看出,一般的隐私保护机制在较强的隐私保护支持下,数据的实用性降低,而本文提出的数据发布算法可以在数据隐私性相同的情况下保持相对较高的数据实用性。

4 结束语

在差分隐私保护下,本文提出了隐私轨迹校准和发布系统,解决了基于噪音前缀树的轨迹数据发布问题,基本上弥补了历史算法中存在的距离局限性以及序列丢失等问题。此方法通过建立噪音增强的前缀树来实现带有隐私保证的嘈杂校准轨迹发布解决方案,可扩展到大型地理空间领域,能够有效保护轨迹数据中的个人隐私信息,提高数据的利用率。然而,当数据集增加到一定限度后隐私预算必定会被耗尽,并且随着数据集的增加,所添加的噪音也会加大,从而影响发布数据的实用性。因此,该算法对

动态轨迹数据发布的隐私保护仍不能完全适用。今后的研究方向将集中在如下 2 个方面：一是如何在不影响数据隐私性和实用性的前提下进一步减小算法的计算量；二是如何在差分隐私保护下实现数据的增量更新。

参考文献

[1] 霍峥，孟小峰，黄毅. PrivateCheckIn：一种移动社交网络中的轨迹隐私保护方法[J]. 计算机学报，2013，36(4)：716-726.

[2] SWEENEY L. k-anonymity：A model for protecting privacy[J]. International Journal of Uncertainty，Fuzziness and Knowlege-Based Systems，2002，10(5)：557-570.

[3] ABUL O，BONCHI F，NANNI M. Never walk alone：Uncertainty for anonymity in moving objects databases[C]//2008 IEEE 24th International Conference on Data Engineering. Cancun，Mexico：IEEE Computer Society，2008：376-385.

[4] 张啸剑，孟小峰. 面向数据发布和分析的差分隐私保护[J]. 计算机学报，2014，37(4)：927-949.

[5] CHEN R，FUNG B C M，DESAI B C，et al. Differentially private transit data publication：A case study on the montreal transportation system[C]//Proc. of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2012). Beijing，China：ACM，2012：213-221.

[6] CHEN Rui，FUNG B C M，DESAI B C. Differentially private trajectory data publication[J]. arXiv preprint arXiv：1112.2020，2011.

[7] CHEN Rui，ACS G，CASTELLUCCIA C. Differentially private sequential data publication via variable-length n-grams[C]// ACM Conference on Computer and Communications Security (CCS). Raleigh，USA：ACM，2012：638-649.

[8] DWORK C，MCSHERRY F，NISSIM K，et al. Calibrating noise to sensitivity in private data analysis[M]//HALEVI S，RABIN T. Theory of Cryptography. TCC 2006. Lecture Notes in Computer Science. Berlin/Heidelberg：Springer，2006，3876：265-284.

[9] 周水庚，李丰，陶宇飞，等. 面向数据库应用的隐私保护研究综述[J]. 计算机学报，2009，32(5)：847-861.

[10] 傅继彬，张啸剑，丁丽萍. MAXGDDP：基于差分隐私的决策数据发布算法[J]. 通信学报，2018，39(3)：136-146.

[11] CHO E，MYERS S A，LESKOVEC J. Friendship and mobility：User movement in location-based social networks[C]// Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego，CA，USA：Dblp，2011：1082-1090.

[12] ESTER M，KRIEGEL H P，SANDER J，et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]//Proceeding of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96). Portland，OR：AAAI，1996：226-231.

[13] LIU Hechen，SCHNEIDER M. Similarity measurement of moving object trajectories[C]// Proceedings of the 3rd ACM Sigspatial International Workshop on GeoStreaming. Redondo Beach，California：ACM，2012：19-22.

[14] 周水庚，范晔，周傲英. 基于数据取样的 DBSCAN 算法[J]. 小型微型计算机系统，2000，21(12)：1270-1274.

(上接第 8 页)

[3] BOYKOV Y，JOLLY M P. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images[C]//2001 IEEE International Conference on Computer Vision (ICCV). British Columbia，Canada：IEEE，2001：105-112.

[4] BERG T，LIU Jiongxin，LEE S W，et al. Birdsnap：Large-scale fine-grained visual categorization of birds[C]//2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Columbus，OH，USA：IEEE，2014：2019-2026.

[5] FELZENSZWALB P F，MCALLESTER D A，RAMANAN D. A discriminatively trained，multiscale，deformable part model[C]// 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Anchorage，Alaska，USA：IEEE，2008：1-8.

[6] BERG T，BELHUMEUR P N. Poof：Part-based one-vs.-one features for fine-grained categorization，face verification，and attribute estimation[C]// 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Portland，OR，USA：IEEE，2013：955-962.

[7] WAH C，BRANSON S，WELINDER P，et al. The caltech-ucsd birds-200-2011 dataset[R]. California：California Institue of Technology，2011.

[8] KHOSLA A.，JAYADEVAPRAKASH N，YAO Bangpeng，et al. Novel dataset for fine-grained image categorization：Stanford dogs[C]// Proceedings CVPR workshop on fine-grained visual categorization (FGVC). Colorado：Spring，2011，2：1-2.

[9] JAIN S D，XIONG Bo，GRAUMAN K. Pixel objectness[J]. arXiv preprint arXiv：1701.05349，2017.

[10] EVERINGHAM M，VAN GOOL L，WILLIAMS C K I，et al. The pascal visual object classes (VOC) challenge[J]. International Journal of Computer Vision，2010，88(2)：303-338.

[11] CSURKA G，DANCE C R，FAN Lixin，et al. Visual categorization with bags of keypoints[C]// Workshop on statistical learning in computer vision，ECCV. Prague：dblp，2004：1-16.