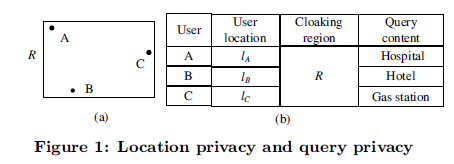
基于失真匿名在连续查询基于位置的移动服务

摘要

隐私保护最近已经受到基于位置的移动服务的相当大的关注。 已经提出了各种位置隐藏方法来保护移动用户的位置隐私。 然而，现有的隐藏方法不适合于连续查询。 鉴于隐私泄露和连续查询匿名化下的QoS（服务质量）差，本文提出了一种δp隐私模型和δq失真模型，以平衡用户隐私和QoS之间的权衡。 此外，两个增量实用程序的隐藏算法 - 自下而上伪装和混合伪装，被建议匿名连续查询。 实验结果验证了所提算法的效率和有效性。

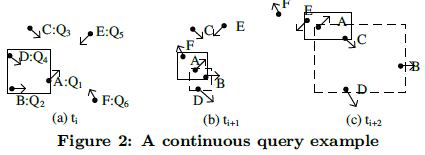
1. 介绍

随着无线通信和移动定位技术的发展，基于位置的服务（LBS）近年来越来越受欢迎。 已经研究努力调查如何保持移动用户的隐私，同时仍然确保LBS的高质量。 通常，存在两种类型的隐私问题：位置隐私[6]（敏感位置被保护以免被链接到特定用户）和查询隐私[4]（查询被保护以免被链接到特定用户）。 例如，假设Alice通过移动电话向服务提供商（SP）（例如GoogleMap）发出以下连续查询：“最近的皮肤病医院在哪里是30分钟？”。 关于位置隐私，Alice想要在她的移动期间隐藏她的确切位置（例如，在诊所或酒馆）; 关于查询隐私，她想隐藏上述关于皮肤病医院的查询是由她发出的事实。



为了保护位置隐私，Gruteser和Grunwald [6]提出了基于位置k匿名模型的时空隐藏，即隐藏位置与至少k-1个其他用户的位置信息不可区分。为了实现位置k匿名性，将每个用户位置扩展到隐藏区域，使得每个区域覆盖至少k个用户。图1（a）示出了位置3-匿名（k = 3）的示例，其中A，B和C的位置被扩展到区域R（即，用户A，B和C形成隐藏集）对手不能知道他们在R的真实位置。在某些情况下，对手知道用户的真实位置[4];因此，查询中包含的位置将成为将标识符链接到特定用户的准标识符（QI）[12]。幸运的是，位置k匿名模型也适用于解决此查询隐私问题。考虑图1（b）所示的示例，通过简单地将查询中包含的位置扩展到相同区域R，可以成功地隐藏精确查询位置，因此查询隐私保存。

大多数现有的隐藏算法专注于匿名快照查询[11,5,7]。 由于相同用户的隐藏集在不同的时间戳[4]是不同的，直接应用这些算法到连续查询不足以保护查询隐私。 图2描绘了在连续查询下公开查询隐私的示例。 如图所示，六个用户A〜F分别发出六个不同的连续查询Q1〜Q6，并且A成功隐藏为a，B，D}，... A，B，F}和.A，E，C} 时间戳ti，ti.1和ti.2。 每个隐藏集合与位置3-匿名一致。 然而，由于它们的交集仅包含A，所以对手可以容易地推断出A发布的查询Q1，并且因此公开了A的查询隐私。



从上述示例可以观察到，隐私泄露是由于在不同时间戳处对相同用户使用不同隐藏集。为了解决这个问题，从相同的隐藏区域发出的查询应该在所有的时间戳[4]。详细地，在这样的方案下，图2中的隐藏集合A在图2中应该总是在t 1到t 2期间保持为A.A，B，D}（隐藏区域在图2（b）和2（c））。虽然该方案成功地保护了查询隐私，但是它导致新的问题：1）可以公开用户位置隐私。如图2（b）所示，.A，B，D}的最小有限矩形（MBR）缩小到ti.1处的较小区域，这可能违反位置隐私要求在最坏的情况下，它可能收缩到一个点，从而暴露真正的用户位置。 2）服务质量变差。如图2（c）所示，大小隐藏区域.A，B，D}在ti.2处显着大，这将使后续的查询处理更加昂贵。在极端情况下，用户可能随着时间的推移分散在整个空间上，迫使隐藏区域覆盖整个区域。

上述新问题背后的原因是，该算法仅利用当前用户位置的接近度，而忽略它们的未来位置。如我们所知，用户的未来位置取决于她的移动的速度和连续查询的持续时间。在理想情况下，相同隐藏集中的所有用户都会移动具有相同的速度，使得隐藏区域的大小在所有时间戳保持相同。不幸的是，这在实践中不太可能发生，并且一旦所涉及的位置更新，位置接近度倾向于改变。具体地，一方面，位置靠近当前时间戳的查询在未来时间戳可能变得彼此远离;另一方面，现在远离彼此的查询可以在某个未来时间戳处满足。对于位置动态变化的连续查询，很难为所有时间戳找到最佳隐藏区域。主要的挑战是如何实现良好的QoS，同时仍然保留查询隐私在查询期间频繁的位置更新。

在本文中，我们考虑对连续查询的位置隐私和查询隐私的保护。 为了解决这个问题，我们提出了δp隐私模型和δqdistortion模型，以平衡用户隐私和QoS之间的权衡。 采用伪装区域的周长来评估位置信息的失真。 如[10]中所指出的，具有类似模式的移动对象最终将在簇中移动。 受此观察的启发，我们提出将位置失真映射到查询的相似性距离，基于哪些查询被聚类，使得每个群集中的位置信息的失真最小化。 当查询移入和移出时，这些集群会逐步维护。

我们在本文中所做的贡献可以总结如下：

* 我们提出了一个p-privacy模型和一个qq失真模型来平衡用户隐私和QoS之间的权衡在连续查询。
* 我们建议将位置失真映射到查询的时间相似性距离。 此外，我们提出两种基于增量效用的掩蔽算法。
* 进行了一系列实验来评估我们提出的算法的性能。 实验结果验证了我们提出的算法的效率和有效性。

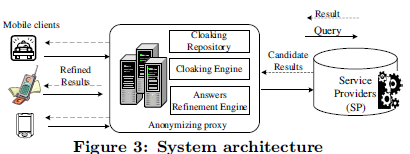
本文的其余部分组织如下。 我们回顾第2节中的相关工作。调查中的问题在第3节中正式定义。几个基于效用的在第4节中提出了隐蔽算法。在第5节中提出了失真和隐私验证的算法。第6节介绍了我们提出的算法的性能评估结果。 最后，本文在第7节结束。

1. 相关工作

位置隐私和查询隐私是涉及基于位置的移动服务的两种类型的隐私问题。 位置k匿名是最流行的位置隐私度量。 这是由Gruteser和Grunwald [6]提出的，后来在[11,1]中被完善。 在用于保护位置隐私的技术方面，现有的方法可以分为隐藏[6]，伪[8]和加密[5]。然而，所有上述工作集中在快照查询的隐私保护。

大多数先前的研究不区分位置隐私和查询隐私。第一个工作，区分他们，并探索隐私保护持续查询是在[4]中提出。尽管如此，它有两个缺点。首先，仅使用发出时间的查询位置来生成隐藏集，这可能导致位置隐私披露和较差的QoS，如上一节所讨论的。其次，由于每个查询的有效期被忽略，连续查询可能被隐藏快照查询。如果任何快照查询移出服务区，则相同隐藏集中的所有连续查询可能不再满足位置k-匿名隐私要求。我们的工作也采用位置k匿名和记忆用户，但它不同于[4]在以下方面。首先，使用时间位置失真模型来找到隐藏集合。其次，具有相似到期时间的查询被聚集在一起，这保证每个连续查询将在其有效期间总是满足隐私要求。

另一个解决连续查询的位置匿名的工作在[15]中给出。 它使用熵来通过假设来测量伪装区域的匿名性水平隐藏区域中的用户的概率不相等。 然而，因为熵不考虑用户位置是否真的不同，所以位置隐私可以在k个不同用户处于相同位置时被公开。 在[13]中，提出了移动感知隐藏算法来保护跟踪分析攻击。 然而，在[13]中使用的隐私度量是位置粒度，而不是本文中考虑的位置k-匿名。



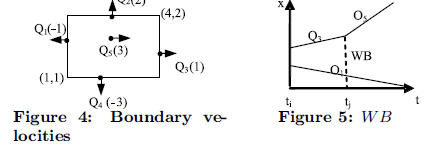
虽然大多数关于LBS中的隐私保护的研究没有讨论位置数据效用，但是其一些测量已经在数据发布中提出，例如泛化高度，可分辨性，信息丢失，分类度量和信息获取隐私损失比[9]。数据发布中的数据实用主要集中在如何保留原始数据的分布用于数据挖掘的目的，而在LBS中，我们主要关注如何广义位置接近原始位置。因此，在本文中，我们采用信息损失，即位置失真作为效用度量。我们注意到我们的位置失真在以下方面不同于[14]。首先，由于数据发布中的QI属性是独立的，每个属性可以与权重相关联以反映其重要性。然而，LBS中的信息，包括位置（x，y）和速度v，彼此相关。时间t。第二，只要给出匿名​​表，在[14]中的信息失真是静态的，而在我们的论文中的位置失真是随时间变化而变化的时间函数。

3.初步

3.1系统架构

像大多数现有的工作[6,11]，我们采用一个集中式系统，其中包括移动用户，可信匿名代理和不受信任的SP，如图3所示。每个移动用户发送基于位置的查询 匿名代理。 有两种类型的查询：新查询和活动查询。 新查询，顾名思义，是用户新发布的查询。 活动查询是在以前某个时间发出但尚未过期的连续查询。 例如，用户在ti发出连续查询Q，其有效期为Δt。 然后，在ti，Q是新查询，而对于任何t∈（ti ti +Δt]，它被认为是一个活动查询。

该匿名代理包括隐形发动机，隐形资源库，并回答细化引擎。当收到一个新的查询，隐形发动机替换为假名ID的用户ID。同时，它调用的位置隐形算法按照生成一个伪装区域用户的隐私要求，这伪装组被保存在（CID展示Qset RL吨RVT）1的形式的隐形库一旦接收到活动查询，伪装引擎搜索原始伪装组，将其生成在发行的时候，在隐形仓库，然后计算出新的隐形地区RLT。后来，匿名代理转发隐形查询到SP，通过维持一个隐形库中，匿名代理可以增量计算隐形集（即，通过为活动查询更新原始伪装区域），并由此获得更高的效率。



最后，SP生成的候选结果首先由回答细化引擎精化，然后中继到移动用户。 在本文中，我们专注于位置掩蔽算法，其考虑位置数据效用以及用户指定的隐私要求。

为了方便我们的研究，我们进一步做出以下假设。 1）每个移动用户都是可信的 - 这是传统位置隐私保护技术中的一个常见假设[6,11,1]。 2）用户在查询期间的移动速度保持不变。 在这个假设下，每个查询的运动函数是线性的。

3.2初步

定义1.（基于位置的查询）查询Q表示为Q =（l ¯v t Texp con），其中（l ¯v t）意味着Q在位置l =（x，y） 在时间戳t处的速度ν=（vx vy），Texp是查询到期时的时间戳，con是该查询的内容。

定义2.（隐藏集）每个隐藏集CS被形式化为：CS =（CID Qset RL t Rv t） 其中CID是这个隐藏集合的标识符，Qset是包含在CS中的查询集合，RL t=（Lx- t Ly- t Lx t Ly t）是位置MBR Qset中的查询，以及Rv t=（vxmin t，vymin t，vxmax t，vymax t）是BVR（边界速度矩形）。 vxmin t= min（vx t，vx- t），vxmax t= max（vx t，vx- t），vymin t= min（vy t，vy- t） ，vymax t= max（vy t vy- t），其中vx- t（vx t）是查询在Lx- t（Lx t）上的边界速度 维度，vy- t（vy t）是Ly- t（Ly t）上的查询在时间t的y维上的边界速度。

请注意，vxmax t（vymax t）和vxmin t（vymin t）可能不是xset（y）维上Qset中的最大和最小速度。 因此，隐藏集的边界上的查询随着用户移动而改变。 因此，Rv t和RL t都是逐段函数w.r.t. t，即：

（Lx- t Ly- t）=（Lx- ti-1 Ly- ti-1）+（vx- t vy- t）×[t-ti-1]

（Lx t Ly t）=（Lx ti-1 Ly ti-1）+（vx t vy t）×[t-ti-1]（1）

其中t∈[ti-1 ti]。 以图4为例，查询Q1〜Q5在时间ti构成隐藏集合CS。 括号中的数字是查询的速度，箭头表示移动方向。 CS.RL ti=（1,1,4,2）和CS.Rv ti=（ - 1 -3 1 2）。 这里，x维上的最大速度为3（对于Q5），但是它不是在时间ti的边界速度。 如果Q5在时间tj超过Q3，则在tj处vxmax将改变为3。

定义3.（边界的宽度/高度）对于具有MBRRL t的隐藏集CS，其在时间t处的宽度，表示为

WBt，WBt =Lx t-Lx- t（2）

类似地，其在时间t的高度，表示为

HBtHBt =Ly t-Ly- t（3）