位置大数据中一种基于 Bloom Filter 的匿名保护方法

刘 彦[1] 张 琳[1,2]

(南京邮电大学计算机学院 南京 210003)[1] (江苏省无线传感网高技术研究重点实验室 南京 210003)[2]

摘 要 位置大数据服务应用中存在大量的用户敏感信息,针对服务中海量数据分析的隐私泄露问题,提出一种基于 Bloom Filter 多哈希散列编码的位置匿名保护方法。采用启发式的隐私度量技术划分匿名区来隐藏真实的位置数据,保持欧氏距离上搜索目标的邻近关系以优化空间匿名框的面积,并在划分策略中引入查询服务相似性因子以减少空间碎片的产生。在移动用户和服务器之间构建可信的第三方位置匿名服务器,能有效地模糊目标节点,从而抵御恶意的隐私攻击。理论分析和仿真实验表明,新算法能有效优化匿名空间区域,提高隐私保护程度,并在海量数据集的构建过程中具有较优的时间复杂度。

关键词 位置大数据服务,隐私保护,位置敏感哈希,匿名区搜索

中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.06.024

Improved Location Anonymous Technology for Big Data Based on Bloom Filter

LIU Yan[1] ZHANG Lin[1,2]

(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)[1]

(Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China)[2]

Abstract As there exists large amounts of user's sensitive information in the application of big data for location, a kind of anonymous location protection method was put forward in this paper, which is based on Bloom Filter with multi-Hash coding, to solve the privacy leakage in analysis of massive data. Heuristic privacy metrology divides anonymous area to hide real data of location. Keeping the search target adjacent in Euclidean distance can optimize the area of spatial anonymous box, and the introduction of similarity factor in query service for dividing policy can reduce space debris. It can effectively blurred target node, with deployment of trusted anonymous server between the mobile user and the server by third-party, to resist malicious privacy attack. Theoretical analysis and simulation results show that the new algorithm can optimize the anonymous space and improve the privacy protection effectively, and it has better time complexity in the construction of massive data sets.

Keywords Location service on big data, Privacy preserving, Locality sensitive hashing, Anonymous spatial region

1 引言

社交网络与移动传感设备等位置感知技术相结合形成了位置大数据服务,面向用户的位置数据分析和发布等应用需求也随之出现和发展,查询业务种类广泛,包括兴趣点 POI (Points of Interest)查找、APP 运动记步、导航定位和社交网络位置分享等功能。由于通信网络中位置大数据服务用途多样,收集信息交叉冗余,如何保护隐私数据和防止用户敏感位置泄露成为当前位置服务面临的重大挑战。对位置大数据服务的恶意访问可能泄露个人大量的敏感信息,随着个人隐私保护意识的不断提高,基于空间匿名区泛化隐藏敏感位置的技术不断出现。

在基于位置的查询过程中,为避免用户主体与行为产生关联,通常采用简单生成的包含目标点的 k 个用户生成匿名集 AS 来申请服务。此类隐私保护技术面临一个基本的计算问题,即最近邻问题[1-3]。文献[4-8]提出基于 k-anonymity 保护计算空间匿名区面积的优化方案,如间隔匿名算法、Casper 匿名以及 Hilbert 匿名等,可以用桶划分泛化查询区域的方式来防止用户身份信息泄露。针对大规模高维数据,改进位置敏感哈希 LSH(Locality Sensitive Hashing)[9-11]是一种快速近似查询的索引技术,具有强大的近邻查找能力,能将高维特征或复杂的距离函数嵌入到一个低维的空间,文献[12]给出了基于欧氏距离上精确位置敏感哈希 E2LSH 的实现,能保证算法以线性时间进行相似性搜索并在实际多种场合中得

到稿日期:2016-04-28 返修日期:2016-09-26 本文受国家自然科学基金(61402241,61572260,61373017,61572261,61472192),江苏省科技支撑计划(BE2015702)资助。

刘 彦(1992—),女,硕士生,主要研究方向为数据挖掘、隐私保护等;张 琳(1980—),女,博士后,副教授,硕士生导师,主要研究方向为云计算、网络安全、信任、可信计算等,E-mail:zhangl@njupt.edu.cn。

到应用[13-17]。在分布式网络中，文献[18-19]提出一种增强型 Bloom Filter 网络流量过滤技术，对通信数据在集合中的映射关系设置了适当清洗并进行可靠的入侵检测，最大限度地减少了安全信息的重建且提高了平均存储命中率。

在多维大数据的复杂搜索中，敏感哈希算法[20-21]修剪子树分支来缩小搜索范围，在保持一定准确率的基础上减少算法空间和时间开销。针对处理海量位置数据集的效率优化问题，本文提出一种基于 Bloom Filter 的快速散列方法，并在移动用户和服务器之间构建一个新的可信第三方服务器模型，完成对目标位置节点的综合匿名处理，合理地分割空间，从而保护敏感位置的隐私信息。大数据环境下，Bloom Filter 数据结构以多哈希位编码的方式对海量数据进行加密，并具有高效的数据搜索、插入、删除效率，仿真实验证明该算法能够在指定的私密性条件下，为位置查询服务返回一定准确率的数据结果。

2　位置大数据匿名保护中的关键问题

2.1　位置隐私的攻击保护模型

随着互联网大数据技术的不断发展，基于位置服务 LBS (Location-Based Services)的网络应用利用全球卫星导航定位系统以及地面通信基站，结合云计算、海量数据处理技术，将数以亿级甚至数以千亿级的位置数据进行实时的挖掘存储。然而，位置大数据应用服务在提供相应查询并发布相关用户信息时，将面临泄露客户个人隐私的危机。恶意攻击针对空间匿名数据的时效性、紧密性等特征进行连续查询攻击，将在某时刻生成过多的空间分割碎片从而导致位置匿名的失败，影响查询效率。为了对发布的位置数据进行处理以降低攻击者推测出用户敏感位置的可能性，在移动用户和服务器之间构建一个可信任的第三方服务器，采用启发式的隐私度量技术将用户的位置数据转换成不真实的位置数据，同时在服务器中考虑优化空间匿名框的面积以及设定碎片分裂的阈值，最后将模糊数据的查询结果转化成用户需要的结果，保护模型如图 1 所示。

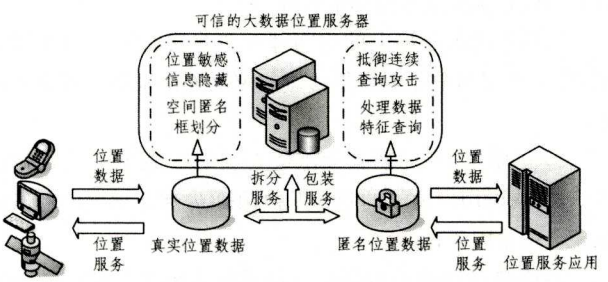

图 1　位置隐私的攻击保护模型

基于空间匿名框的位置大数据隐私保护技术假设用户在任意时刻 t 发布位置信息后，暴露的用户敏感信息只与当前时刻攻击者收集到的数据有关。攻击者可以根据在任意时刻 t 之前收集到的历史位置数据 L，推测用户在 t 时刻处于某个敏感位置 l_n 的概率为 $P(l_n|L)$。为了量化位置大数据的隐私保护效果，对位置大数据的 θ 隐私定义如下。

定义 1　设任意用户 U 在任意时刻 t 处于位置 l_n 的推测概率为 $P(U_{l_n}^t)$，在 t 时刻之前收集到关于用户 U 的历史位置数据为 $L^t=\{l_1^t,l_2^t,\cdots,l_{n-1}^t\}$，则

$$P(U_{l_n}^t|L^t)-P(U_{l_n}^t)\leqslant\theta \tag{1}$$

其中，θ 是用户给定的隐私需求，也是攻击者能够获得的最大攻击效果。攻击者根据 t 时刻之前获取到的历史位置数据，推测用户在 t 时刻处于位置 l_n 的后验概率 $P(U_{l_n}^t|L^t)$ 与其先验概率之差不能超过隐私需求 θ，从而量化了隐私保护效果，当 $\theta=0$ 时隐私保护效果达到最大，称为完美隐私。

当匿名发布的假位置数据可用性低于用户的查询服务要求时，可以根据应用的查询函数特征对可信服务器反馈误差信息，在隐私保护范围内要求适当降低空间泛化程度来提高位置数据的可用性。位置服务中查询策略的相关度量方式的定义如下。

定义 2　设 $G(Q)$ 为原始数据的查询结果，$G^{\wedge}(Q)$ 为匿名后数据的查询结果，则近似度 S_Q 可定义为两输出结果的一阶范数距离 $S_Q=\|G(Q)-G^{\wedge}(Q)\|_1$。对连续查询 $Q_i\in\{Q_1,\cdots,Q_k\}$，位置服务发布数据的可用性定义为：

$$U=\frac{1}{|Q_i|}\sum_{Q_i}(S_{Q_i}/e^{\varepsilon/2}) \tag{2}$$

服务对象定义允许的误差范围为 δ，则可容忍的信息丢失率为 $1-\delta$，误差范围 δ 与数据可用性 U 成反比。其中，基于空间距离的查询函数敏感度为 $\sigma(f)=1$，而针对海量位置数据集中的一组连续分区查询策略 $F_v=\{f_p:D\to(R^+)^d\}$，定义其全局敏感度为其中任意查询函数敏感度的最大值，表示为 $\sigma(F_v)=\sup_{i\in v}\sigma(f_i)$。

2.2　位置大数据的空间模糊化

常见空间模糊化技术采用某种划分策略将二维空间位置数据集 S 划分成不相交的子集或桶，桶中用户满足共匿性要求并保持欧氏距离上的 K 邻近。匿名器将每个桶中的用户的位置坐标泛化成一个具有 k-匿名性质的区域，即匿名区 ASR(Anonymous Spatial Region)，用户敏感信息在该区域内被识别出来的概率为 $1/k$。在不同稀疏密度的位置空间分布中，目标位置 d 的桶分配、桶合并策略将与算法的复杂程度以及效率紧耦合。文献[4]提出的间隔匿名(interval cloak)算法中，虽然邻近的用户划分满足位置的共匿性要求，但其递归过程有两个缺点：1)分在同一个桶中的用户在原始的数据集中可能并没有邻近关系，尤其对于 k 值较大的情况容易产生空间碎片；2)该算法的查询函数不具备保距性，将产生较高的时间复杂度 $O(\frac{n^2}{k})\log(\frac{n^2}{k})$。

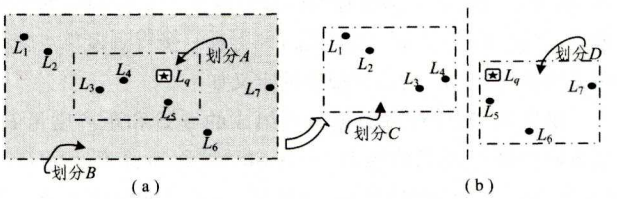

图 2　简单最近相邻空间分割问题

在将目标位置点汇聚成桶匿名的处理过程中，基于最近邻分区的自然分裂策略根据最近距离选择划分区域，在此位

置桶的划分方式中较易产生空间碎片，消耗更多的系统资源。如图 2 所示，首先选择目标位置点 L_q 并假设此时设置的空间泛化匿名度为 $k=4$。计算与 L_q 距离最近的 3 个邻接点形成的匿名区域 ASR，如图 2(a)中划分 A 的矩形区域所示。将包含目标位置的 4 个点移出数据集后，剩余点组成另一个匿名区域 B(阴影部分)，可以观察到这种桶划分后的空白区域较大，易产生空间碎片，显然图 2(b)所示的空间分割方式的位置点更汇聚，更为可取。

在位置服务过程中为提高空间划分的合理性，不仅要考虑空间匿名的保距性，还要考虑查询服务的相似性。可信的第三方服务器在划分策略中引入基于查询结果的位置服务近似度因子，根据查询返回结果实时反馈算法性能，设置阈值控制 ASR 面积，对空间点泛化匿名程度进行适当的调节，能给出更加聚集的空间点划分区域，并提高位置服务的精确性。在敏感位置点与相应查询结果的映射关系中，利用多个哈希函数同时考虑近似查询与间隔距离参数，综合评估位置点的性质对敏感位置进行模糊化，从而满足用户更高的隐私保护需求 θ。其位置服务的相似度 S 定义如下。

定义 3 假设对两个不同的位置 p 和 q 进行查询服务时，其查询结果将具有一定的相似性，在查询函数 Q 的查询结果集 $G(Q)$ 中，对用户 *top-k* 个兴趣点的排序结果集表示为 $G_k(Q)$，目标位置根据相应空间坐标 (x,y) 标记，对查询结果的处理可以计算出其位置的服务相似度为 $S(0\leqslant S\leqslant 1)$。

$$S=Sim(p,q)=Sim((x_p,y_p),(x_q,y_q))=\frac{G_k(Q\langle x_p,y_p\rangle)\bigcup G_k(Q\langle x_q,y_q\rangle)}{k} \tag{3}$$

2.3 位置敏感的哈希过滤器

位置敏感哈希 LSH 是基于哈希数据结构的随机映射算法，其基本思想是保证以较大的概率将空间上相距很近的点进行随机映射存储操作后，仍保持这些位置点距离上的邻近关系。空间划分映射关系存储在哈希表中，可以使一个目标点的查询在 O(1)的时间复杂度内以及 O(n)的内存空间上完成。布隆过滤器 Bloom Filter 基于 m 个 Hash 函数将一个元素映射成一个位阵列(bit array)存取，是一种对海量数据处理效率很高的随机数据结构。元素加入集合时，计算 m 个独立 Hash 函数并将其在位阵列对应值置 1 或 0，查找该元素时若搜索位数组中对应位值都是 1，则说明存在。空间中的相邻点对于每个哈希函数发生冲突的概率比距离远的点要大，能将比较相近的点哈希到同一个桶中。可选择欧氏距离度量、服务近似度以及添加查询函数组成多个哈希函数的影响因子，并获取它所在桶中的标志，即可进一步得到聚集点较强相关的空间划分。对位置敏感的定义如下。

定义 4 对于位置数据节点组成的域 S 和距离度量 D，从 S 映射到 U 的函数族 $H\{h:S\to U\}$ 被称为关于度量 D 是 (r_1,r_2,p_1,p_2) 位置敏感的，如果 $\forall v,q\in S$ 满足以下两个条件：

(1)如果 $d(v,q)<r_1$，那么 $P_H[h(q)=h(v)]\geqslant p_1$；

(2)如果 $d(v,q)>r_2$，那么 $P_H[h(q)=h(v)]\leqslant p_2$。

其中，$r_1<r_2$，$p_1>p_2$，$d(v,q)$ 是 D 中 v 与 q 的距离。如果位置查询点 q 与 v 邻近，那么哈希方法 h 散列后划分到同一桶中的概率增大。当 q 与 v 存在其他如查询服务的近似关系但彼此不够邻近时，需要多个哈希函数连接形成划分策略 $H(h_1(v),h_2(v),\cdots,h_m(v))$ 放大空间点的关联关系，其中的哈希函数簇 $h_i\in H$ 相互独立。基于 Bloom Filter 多哈希快速索引思想组合位置服务数据中的多种影响关联因子，同时保持空间上的聚合关系完成空间点集的桶划分，能得到更为合理、更满足实际需求的查询结果。

3 基于 Bloom Filter 的海量位置信息保护技术

3.1 海量位置点匿名保护方法

现有的位置大数据服务中，攻击者可以从多种渠道获得用户和位置数据相关的其他类型数据，并结合位置数据共同推测用户的隐私信息。如许多社交软件的签到服务、分享定位等操作，通过用户的个性设置或者行为模式对其位置数据与非位置数据进行匹配，会将用户大量敏感的个人信息发布给基于位置服务的攻击者，带来隐私泄露的潜在危机。常用的位置隐私保护技术中普遍存在两个缺陷：1)基于启发式隐私度量的空间模糊化方法不能生成精确合理的匿名区 ASR，较易生成空间碎片从而造成算法的冗余；2)大多数匿名技术适用的数据量较小，针对位置大数据服务中海量关联数据的处理方法效率不高。

为解决上述问题，本文提出构建基于含有安全客户信任中间件的匿名保护系统，位置敏感哈希算法采用启发式的隐私度量技术划分匿名区域，并保持欧氏距离上搜索目标的邻近性隐藏真实位置数据，以达到隐私信息安全通信的目的。通过中间件反馈服务结果调整匿名区域的面积大小并平衡分裂聚合条件，再利用 Bloom Filter 结构完成对海量数据集的高效搜索，其中多个哈希函数组成的阵列将直接影响匿名框的覆盖率和隐私性，从而最终影响位置大数据服务器对隐私位置数据安全发布的效率。本文提出基于 Bloom Filter 的海量位置信息保护技术的基本流程如下：

1)基于欧氏距离上的空间邻近关系递归构建四叉树索引，将二维空间中位置点集自然的位置关系映射在树形结构中；

2)根据用户所需的查询函数 Q 计算位置服务的近似因子 S，并获取用户要求的隐私保护预算 θ，准备划分匿名区域的距离度量因子；

3)选择 m 个哈希函数组成敏感位置匿名保护的划分策略 $H(h_1(v),h_2(v),\cdots,h_m(v))$，每个哈希函数引入不同的散列度量且彼此保持独立，在用户得到相同位置服务性能的条件下隐藏隐私位置信息；

4)计算哈希函数 h_i 的值，并把结果映射到 Bloom Filter 过滤器的位数组相应位中；

5)根据用户选择的泛化程度搜索 Bloom Filter 中保存的位置点，将目标点和散列后与其依然保持邻近关系的点聚集成桶，由形成的匿名区向应用服务器申请服务，并得到查询结

果反馈给目标点。

可信服务器首先基于四叉树 Quad-Tree 将二维空间递归地十字分成 4 个面积相等的正方形区间,要求系统设定递归分裂的终止条件形成最小的正方形区间保存在树的叶子节点中,每一个划分正方形区域对应于四叉树中的一个节点。当对目标敏感位置点匿名隐藏时,可根据 Bloom Filter 生成的哈希随机散列策略搜索匿名框,统计不同层次的树节点对应正方形区域所包含的位置点并进行桶划分,位置点基于 Bloom Filter 的散列原理如图 3 所示。

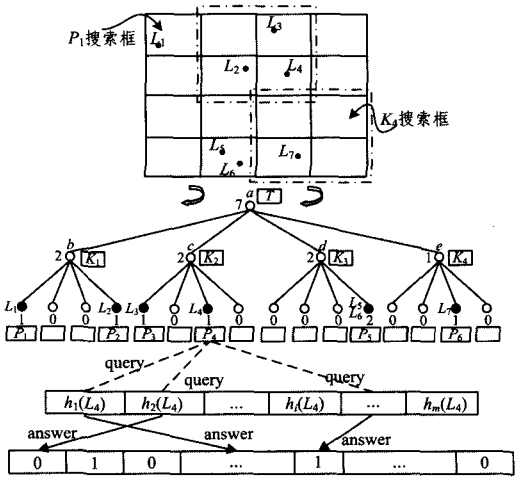

图 3　基于 Bloom Filter 的位置散列原理

由于空间数据的特殊性(海量、多维、空间拓扑特征、时间特征),空间分解技术采用分割原理,在把查询空间划分为若干区域的同时将位置信息分层储存,形成可唯一标识空间的要素。然后利用不同的数据结构对分割的区域进行组织,以达到快速访问数据项的目的。常用的空间索引技术可以提高空间信息数据库的操作效率,Quad-Tree 将位置空间递归划分为不同层次的树结构,当空间数据对象分布比较均匀时,具有比较高的空间数据插入和查询效率;R 树广泛应用于原型研究和商业应用中,为了使 R 树能在海量的空间数据库中发挥重要作用,将优化 R 树节点的插入路径,减少目标空间区域的重叠面积,同时保持当前层节点分裂的独立性;kd_Tree 索引用于多维检索的结构形式,对数据点在 k 维空间中进行划分,并在每一层都根据该层的分辨器对相应对象做出分枝决策,对于精确的点匹配查找具有与二叉树一样的良好性能(平均查找长度为 $1+4\log n$)。

3.2　性能分析

Bloom filter 以内存空间换取时间的处理方式体现了对海量数据的处理优势,在增加或查找集合元素时所用的算法时间复杂度都为哈希函数的计算时间。同时,由于 Bloom Filter 对加入位向量集合中的元素进行了编码,并且从这些编码中恢复出原集合元素信息的过程并不容易,因此可用这种加密处理方式对敏感位置的隐私提供有效保护。Bloom Filter 搜索采用近似邻近的方法,能以较低的代价返回精确的或接近精确的结果,但其在判断一个元素是否属于其表示的集合时存在一定的错误率,所以不能适用于要求零错误率的应用场合。为了优化算法性能对哈希冲突元素进行合理的分配,需分析位置映射的位向量中某位标记改动的随机概率,以及预估计 Bloom Filter 的错误率大小从而最优化策略簇中哈希函数的个数,使处理器在一定容错率的条件下均衡效率。

假设哈希策略 $H(h_1(v),h_2(v),\cdots,h_k(v))$ 中各个哈希函数是完全随机和独立的,当数据集 $S=\{x_1,x_1,\cdots,x_n\}$ 中所有元素都被 k 个哈希函数映射到 m 位的位数组中时,其中 $k\times n<m$,则这个位数组中某一位仍然为 0 的概率 p 为:

$$p=(1-\frac{1}{m})^{kn}\approx e^{-kn/m} \tag{4}$$

其中,$\frac{1}{m}$ 表示任意一个哈希函数选中这一位的概率,$(1-\frac{1}{m})$ 表示哈希从未选中这一位的概率,要把 S 完全映射到位数组中,需要做 kn 次哈希,则某位值为 0 代表 kn 次哈希都未选中。为了简化运算,用极限思想 $\lim_{x\to\infty}(1-\frac{1}{x})^{-x}=e$ 近似,则错误率 f_p 为:

$$f_p=[1-(1-\frac{1}{m})^{kn}]^k\approx(1-p)^k$$

$$=e^{k\ln(1-p)}=e^{-\frac{m}{n}\ln(p)\ln(1-p)} \tag{5}$$

用错误率最优化哈希函数个数,能使元素查询时的错误率降到最低,这里有两个互斥的理由:1)如果哈希函数的个数多,那么在对一个不属于集合的元素进行查询时得到 0 的概率就大;2)如果哈希函数的个数少,那么位数组中的 0 就多。当 m 和 n 的比值越大,则要求哈希函数的个数 k 越大,并且其错误率越小。在式(5)中令 $g=-(\frac{m}{n})\ln(p)\ln(1-p)$,由对称法则可知当 $k=\ln2(\frac{m}{n})$ 即 $p=\frac{1}{2}$ 时,g 取最小值,则 f_p 取得最小值。在不超过一定错误率的情况下,Bloom Filter 至少需要 m 位才能完全表示全集中任意 n 个元素的集合。假设全集中共有 u 个元素,允许的最大错误率为 ε,则 m 至少要等于 $n\log2(1/\varepsilon)$ 才能表示任意 n 个元素的集合,即满足以下不等式:

$$m\geqslant\log_2\frac{\binom{u}{n}}{\binom{n+\varepsilon(u-n)}{n}}\approx\log_2\frac{\binom{u}{n}}{\binom{\varepsilon u}{n}}\geqslant\log_2\varepsilon^{-n}$$

$$=n\log_2(1/\varepsilon) \tag{6}$$

4　实验结果及分析

实验数据源采用 UCI[22] 机器学习数据库中关于智能手机运动传感器读取的样本数据集(Heterogeneity Activity Recognition),实验系统硬件配置为主频 2.4GHz 的 Intel(R) Core(TM)i3 兼容 PC 机,2GB 内存,200G 以上的可用磁盘空间,软件配置平台为 WIN 2008 Server 操作系统以及 Microsoft SQL Server 数据库系统,C/S 结构的运行模式。传感器包含了约 3.5G 大小的位置坐标数据集,实验对比基于四叉树索引的 Casper 匿名[4]算法、假位置选择 DLS[5]算法以及 Hilbert[7] 匿名算法,对本文提出的 Bloom Filter 改进匿名保

护方法进行性能评估。其中，采用 ASR 的面积与整个数据空间的比值平均值来度量有效性，采用 ASR 的平均生成时间来度量效率，实验结果如图 4、图 5 所示。

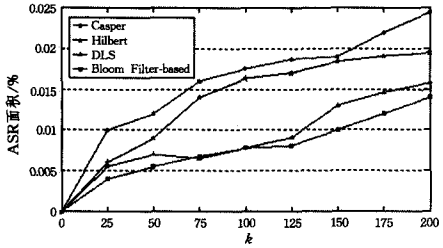

图 4 ASR 的面积与 k 的关系

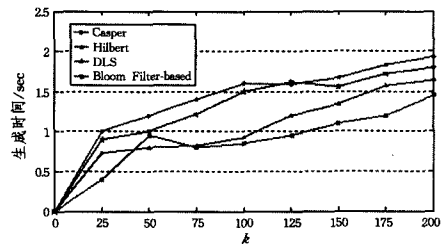

图 5 ASR 生成时间与 k 的关系

用户位置随机分布在 10000×10000 的区域内，在 Bloom Filter 选择 $m=20$ 哈希函数策略族的情况下，图 4 示出了 4 种匿名算法生成的 ASR 与整个面积的百分比与 k 值的关系。实验结果显示，ASR 尺寸基本上随 k 值线性变化，并且基于 Bloom Filter 的匿名方式能给出更为聚集的区域划分。同时，从图 5 的对比可以看出，在大数据环境下布隆过滤器具有较高的 ASR 生成效率。需要指出的是当 $25\leqslant k\leqslant 75$ 时，本文算法生成匿名区的时间有明显的跳跃。这是因为对于不同的匿名保护要求，以 Bloom Filter 位编码为基础的元素搜索方法将产生一定的错误率 f_p，从而处理时间也会相应增加。当 $k\geqslant 75$ 时，随着哈希函数个数 k 的增大，其产生的错误率减小且趋向稳定。在位置敏感哈希分割空间实现共匿性的基础上，新算法具有综合服务近似度查询的优势，并具有适度的计算复杂度，实验证明了该算法在有效性（最小化匿名空间区域）和效率（海量数据集构建代价）方面具有良好的性能。

面向单一攻击的位置数据服务框架已经无法适应多组合复杂攻击的大数据环境。一方面，大部分位置数据发布框架设计忽略了真实空间数据具有数据特征以及动态位置查询环境下服务的相似性，单一考虑位置数据而忽略了非位置数据的隐私保护，无法估计数据多次发布所带来的潜在风险。另一方面，现有位置匿名模型无法兼顾数据安全性与可用性从而及时对用户的反馈做出调整，未能高效地添加适当噪音来减少匿名泛化过程中带来的误差，以提高数据的可用性。本文算法在实验过程中表现的稳定性有效解决了以上问题，在未来大数据位置服务的隐私保护中具有良好的可行性。

结束语 大多数已有的基于位置匿名的隐私保护技术不能综合考虑集群点之间的近似性，在对邻近点的搜索策略中忽略了查询服务的相似关系，单纯考虑距离上邻近的方式将可能产生不合理的匿名空间划分，从而降低服务器的执行效率。另一方面，大数据时代下针对海量信息的分析方法层出不穷，位置应用服务在大数据环境下发展迅速，选择能高效处理数据并保护敏感位置节点信息的数据结构将具有广泛的应用前景。针对以上问题，本文运用 Bloom Filter 多哈希位编码的原理并改进了形似性的评价方式，采用综合的搜索策略加密目标点来达到隐私保护的目的，实验表明本文设计方案具有良好的有效性和运算速度。为进一步提高算法性能，本文后续将继续深入探索基于不同挖掘算法的位置数据安全访问技术。

参 考 文 献

[1] INDYK P, MOTWANI R. Approximate nearest neighbors: towards removing the curse of dimensionality[C]// Proceedings of the 1998 30th Annual ACM Symposium on Theory of Computing. Dallas USA, 1998: 604-613.

[2] ZHANG X J, GUI X L, WU Z D. Privacy preservation for location-based services: A survey[J]. Journal of Software, 2015, 26(9): 2373-2395. (in Chinese)
张学军，桂小林，伍忠东. 位置服务隐私保护研究综述[J]. 软件学报，2015，26(9)：2373-2395.

[3] WANG L, MENG X F. Location privacy preservation in big data era: A survey[J]. Journal of Software, 2014, 25(4): 693-712. (in Chinese)
王璐，孟小峰. 位置大数据隐私保护研究综述[J]. 软件学报，2014，25(4)：693-712.

[4] MOKBEL M F, CHOW C Y, AREF W G. The New Casper: A Privacy-Aware Location-Based Database Server[C]// IEEE 23rd International Conference on Data Engineering, 2007 (ICDE 2007). IEEE, 2007: 1499-1500.

[5] NIU B, LI Q, ZHU X, et al. Achieving k-anonymity in privacy-aware location-based services[C]// IEEE INFOCOM2014-IEEE Conference on Computer Communications. IEEE, 2014: 754-762.

[6] CHOW C Y, MOKBEL M F, AREF W G. Casper*: Query processing for location services without compromising privacy[C]// Proceedings of the 32nd International Conference on Very Large Data Bases. VLDB Endowment, 2006: 763-774.

[7] KRISHNAMACHARI B, GHINITA G, KALNIS P. Privacy-Preserving Publication of User Locations in the Proximity of Sensitive Sites[C]// Scientific and Statistical Database Management, International Conference, SSDBM 2008. Hong Kong, China, 2008: 95-113.

[8] KALNIS P, GHINITA G, MOURATIDIS K, et al. Preventing Location-Based Identity Inference in Anonymous Spatial Queries [J]. IEEE Transactions on Knowledge & Data Engineering, 2008, 19(12): 1719-1733.

[9] QIAN J, ZHU Q, CHEN H. Multi-Granularity Locality-Sensitive Bloom Filter[J]. IEEE Transactions on Computers, 2015, 64(12): 3500-3514.

[10] SLANEY M, CASEY M. Locality-Sensitive Hashing for Finding Nearest Neighbors[C]// IEEE Signal Processing Magazine. 2008: 128-131.

[11] HOU S J,ZHANG Y J,LIU G H. Spatial K-Anonymity Reciprocal Algorithm Based on Locality-sensitive Hashing Partition [J]. Computer Science,2013,40(8):115-118. (in Chinese)
侯士江,张玉江,刘国华. 基于位置敏感哈希分割的空间 K-匿名共匿算法[J]. 计算机科学,2013,40(8):115-118.

[12] DATAR M,IMMORLICA N,INDYK P,et al. Locality-sensitive hashing scheme based on p-stable distributions[C]// Twentieth Symposium on Computational Geometry. 2004:253-262.

[13] KULIS B,GRAUMAN K. Kernelized Locality-Sensitive Hashing[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence,2012,34(6):1092-1104.

[14] FISICHELLA M,DENG F,NEJDL W. Efficient Incremental Near Duplicate Detection Based on Locality Sensitive Hashing [M]// Database and Expert Systems Applications. Springer Berlin Heidelberg,2010:152-166.

[15] LI H M,HAO W N,CHEN G. Collaborative filtering recommendation algorithm based on exact Euclidean locality-sensitive hashing[J]. Journal of Computer Applications,2014,34(12):3481-3486. (in Chinese)
李红梅,郝文宁,陈刚. 基于精确欧氏局部敏感哈希的协同过滤推荐算法[J]. 计算机应用,2014,34(12):3481-3486.

[16] LI H M,HAO W N,CHEN G. Collaborative Filtering Recommendation Algorithm Based on Improved Locality-sensitive Hashing[J]. Computer Science,2015,42(10):256-261. (in Chinese)
李红梅,郝文宁,陈刚. 基于改进 LSH 的协同过滤推荐算法[J]. 计算机科学,2015,42(10):256-261.

[17] LIU Z,LIU T,GIBBON D C,et al. Effective and scalable video copy detection[C]// ACM Sigmm International Conference on Multimedia Information Retrieval. Mir 2010,Philadelphia,Pennsylvania,Vsa,March. DBLP,2010:119-128.

[18] SARAVANAN K,SENTHILKUMAR A. Security Enhancement in Distributed Networks Using Link-Based Mapping Scheme for Network Intrusion Detection with Enhanced Bloom Filter [J]. Wireless Personal Communications,2015,84(2):821-839.

[19] MALHI A,BATRA S. Privacy-preserving authentication framework using bloom filter for secure vehicular communications [J]. International Journal of Information Security,2015,13(1):1-21.

[20] HUO Z,XIAO L,ZHONG Q,et al. MBFS:a parallel metadata search method based on Bloomfilters using MapReduce for large-scale file systems[J]. The Journal of Supercomputing,2016:12(8):3006-3032.

[21] BHUSHAN M,SINGH M,YADAV S K. Big data query optimization by using Locality Sensitive Bloom Filter[C]// International Conference on Computing for Sustainable Global Development. IEEE,2015:70-71.

[22] UCI Machine Learning Repository[OL]. http://archive.ics.uci.edu/ml.

(上接第 143 页)

[9] LI C,HAY M,RASTOGI V,et al. Optimizing Linear Counting Queries under Differential Privacy[C]// Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. 2010:123-134.

[10] XIONG P,ZHU T,NIU W,et al. A Differentially Private Algorithm for Location Data Release[J]. Knowledge and Information Systems,2016,47(3):647-669.

[11] XIAO X,BENDER G,HAY M,et al. iReduct:Differential Privacy with Reduced Relative Errors[C]// Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data. 2011:229-240.

[12] CHAUDHURI K,MONTELEONI C,SARWATE A D. Differentially Private Empirical Risk Minimization[J]. The Journal of Machine Learning Research,2011,12:1069-1109.

[13] ZHANG J,XIAO X,YANG Y,et al. PrivGene:Differentially Private Model Fitting using Genetic Algorithms[C]// Proceedings of the 2013 ACM SIGMOD International Conference on Mana-gement of Data. 2013:665-676.

[14] BLUM A,DWORK C,MCSHERRY F,et al. Practical privacy:the SuLQ Framework[C]// Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. 2005:128-138.

[15] MCSHERRY F D. Privacy Integrated Queries:an Extensible Platform for Privacy-preserving Data Analysis[C]// Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data. 2009:19-30.

[16] FRIEDMAN A,SCHUSTER A. Data mining with Differential Privacy[C]// Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2010:493-502.

[17] JAGANNATHAN G,PILLAIPAKKAMNATT K,WRIGHT R N. A Practical Differentially Private Random Decision Tree Classifier[C]// IEEE International Conference on Data Mining Workshops,2009(ICDMW'09). 2009:114-121.

[18] XIONG P,ZHU T Q,JING D W. Different Private Data Publishing Algorithm for Building Decision Tree[J]. Application Research Computers,2014,31(10):3108-3112. (in Chinese)
熊平,朱天清,金大卫. 一种面向决策树构建的差分隐私保护算法[J]. 计算机应用研究,2014,31(10):3108-3112.

[19] DWORK C. The promise of Differential Privacy:A Tutorial on Algorithmic Techniques[C]// Foundations of Computer Science (FOCS). 2011:1-2.

[20] DWORK C. Differential Privacy in New Settings[C]// SODA. 2010:174-183.

[21] DWORK C. Differential Privacy:A survey of Results[M]// Theory and Applications of Models of Computation. Springer,2008:1-19.

[22] MCSHERRY F,TALWAR K. Mechanism Design via Differential Privacy[C]// 48th Annual IEEE Symposium on Foundations of Computer Science,2007(FOCS'07). 2007:94-103.