一种隐私保护的在线相似轨迹挖掘方法

赵家石，杨 静，张健沛

（哈尔滨工程大学 计算机科学与技术学院，150001 哈尔滨）

摘 要：为了解决相似轨迹挖掘中的隐私保护、轨迹数据简化和在线处理问题，提出了一种能够保护用户原始轨迹数据隐私的在线挖掘相似轨迹的方法.该方法首先利用随机投影技术压缩和扰动原始轨迹数据，然后通过基于密度的聚类方法判定各个时间段内相似的移动对象，采用局部敏感哈希技术寻找在足够多的时间段内都相似的移动对象，避免了传统方法中的交集运算，实现快速估计轨迹间相似度.实验结果表明：该方法能够有效的发现相似轨迹，并且时间开销较小.

关键词：数据挖掘；隐私保护；相似模式；轨迹数据

中图分类号：TP391 **文献标志码**：A **文章编号**：0367-6234(2013)11-0101-05

Privacy aware online mining of similar trajectories

ZHAO Jiashi, YANG Jing, ZHANG Jianpei

(Collage of Computer Science and Technology, Harbin Engineering University, 150001 Harbin, China)

Abstract: The problems of privacy preserving, trajectory data simplification and online processing attract considerable efforts from researchers in the area of trajectory data mining, unfortunately, it is difficult for traditional method to solve all these problems. This paper proposes an online similar trajectories mining method which can preserve the privacy of original trajectory data. The method first compressed and perturbed the original data based on random projection technique, then found the similar moving objects in each time segment by clustering the transformed data based on density, and finally it found similar trajectories by estimating that if the trajectories were similar for a long enough duration and estimated the similarity of trajectories using local sensitivity hashing. This avoided the intersection operation in traditional method and reduced the computation time. The experimental results show that this method can find similar trajectories effectively and reduce the cost of computation time.

Key words: data mining; privacy preserving; similar pattern; trajectory data

近年来随着具备 GPS 定位服务的移动设备的普及，轨迹数据挖掘受到了越来越多的关注[1].其中，针对轨迹数据相似模式挖掘的研究较为广泛[2].轨迹数据相似模式挖掘是将某段时间间隔内的相似的移动对象进行分组以揭示其潜在的相关性，在车辆交通管理、物流管理、模式识别以及社交网络等方面具有巨大的应用价值.目前，相关研究主要解决如下问题：1)如何实现轨迹数据流的在线相似模式挖掘[3-4]；2)如何通过压缩或者简化轨迹数据进行有效的挖掘[5-6]；3)如何在挖掘的过程中不暴露原始的轨迹数据，保护用户隐私[7-8].但是，在现有的研究中缺乏能够同时解决这 3 方面问题的方法.本文提出了一种能够保护原始数据隐私的在线挖掘相似轨迹的方法 PSimTra(Privacy-aware Similar Trajectory).首先利用随机投影技术将原始数据进行转换，实现轨迹数据的压缩和扰动，然后通过基于密度的聚类方法寻找持续时间段内呈聚类状态的轨迹数据集.采用局部敏感哈希(Locality Sensitive Hashing, LSH)技术建立索引进而快速选择相似轨迹数据，

收稿日期：2012-09-09.

基金项目：国家自然科学基金资助项目（61370083，61073043，61073041）.

作者简介：赵家石(1985—)，女，博士研究生；
杨 静(1962—)，女，教授，博士生导师；
张健沛(1956—)，男，教授，博士生导师.

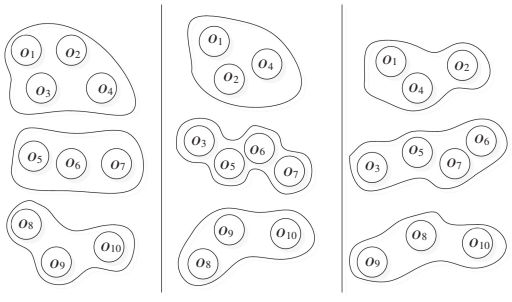
通信作者：赵家石，zhaojiashib09@hrbeu.edu.cn.

最终实现在线挖掘相似轨迹数据.

1 问题定义

令 $O=\{o_1,o_2,\cdots,o_n\}$ 为移动对象的集合,其中 o_i 为移动对象;令 $T=\{t_1,t_2,\cdots,t_m\}$ 为时间域,其中 t_i 为时间戳;移动对象 o 在时间间隔 $[t_1,t_n]$ 内的轨迹表示为 $\mathrm{tr}_o=\{(p_1,t_1),(p_2,t_2),\cdots,(p_n,t_n)\}$,其中 p_i 为 t_i 时刻移动对象 o 的位置.

相似轨迹挖掘问题为:一些移动对象向服务端发送轨迹数据,服务端能够发现某一时间段内持续呈聚集状态的移动对象集合. 如图 1 所示,10 个移动对象分别在 t_1、t_2 和 t_3 时刻位置的聚集状态.相似轨迹是指在连续的一段时间内的各个时刻都呈聚集状态的移动对象集合,图 1 中 $\{o_1,o_2,o_4\}$、$\{o_5,o_6,o_7\}$ 以及 $\{o_8,o_9,o_{10}\}$ 分别为相似轨迹.



(a) t_1 (b) t_2 (c) t_3

图 1 相似轨迹示例

隐私保护问题为:包含大量用户位置信息的轨迹数据发送给不可信的服务提供方而导致的用户隐私泄露.隐私保护的轨迹数据挖掘是通过扰动或匿名等手段将原始数据转换后提供给服务方,服务方可以使用转换后的数据进行挖掘而不影响挖掘结果的准确性.

轨迹数据简化是为了有效的存储、传输和处理大量轨迹数据. 令时间间隔 $[t_1,t_n]$ 内的轨迹数据 $\mathrm{tr}=\{(p_1,t_1),(p_2,t_2),\cdots,(p_n,t_n)\}$ 经过压缩简化为 $\mathrm{tr}'=\{(q_1,t_1),(q_2,t_2),\cdots,(q_m,t_m)\}$,$m<n$,并且在 tr' 上进行的计算结果与在 tr 上计算结果相等或近似,使得原有的时间和空间信息得以保持.

在线处理是指在实时系统中,服务端接收到的轨迹通常是以数据流的形式持续不断的到来,这就需要挖掘方法能够在线实时的更新并处理轨迹数据.因此,在线相似轨迹挖掘方法需要满足以较低的维护成本动态地处理轨迹数据流.

2 在线相似轨迹挖掘

2.1 轨迹数据转换

轨迹数据在用户端发送给服务端之前进行转换,转换过程分为坐标转换、分段和投影转换.

移动对象通过 GPS 设备获取的位置数据通常表示为纬度和经度信息,令移动对象 o 在 t 时刻的位置为 $\mathrm{loc}=(\mathrm{lat},\mathrm{lon},t)$,其中:lat 为 t 时刻移动对象所在位置的纬度;lon 为经度.由于在以后的相似性度量中将采用欧式距离,所以首先对原始位置数据进行墨卡托投影转换.将纬度 lat 和经度 lon 转换为坐标值 x 和 y.

用户端每隔一定的时间段向服务端发送一次位置数据信息.该位置数据信息为该段时间间隔内的轨迹数据.因此需要对时间间隔进行统一界定,进行分段处理.定义每个时间间隔为一个时间桶 T_{bin}:$T_{\mathrm{bin}}=\{t_i,t_{i+1},\cdots,t_{i+d-1}\}$.每个时间桶包含 d 个时间点的位置数据,这 d 个时间点的位置数据组成了该时间桶内的轨迹数据$(\boldsymbol{X},\boldsymbol{Y},T_{\mathrm{bin}})$,其中 $\boldsymbol{X}=(x_1,x_2,\cdots,x_d)$,$\boldsymbol{Y}=(y_1,y_2,\cdots,y_d)$.

为了简化和保护原始轨迹数据,用户端在发送每个时间桶内的轨迹数据之前对其进行随机投影转换.随机投影转换是指将高维空间中的数据点投影到低维空间中,使得任意两点之间的欧式距离近似保持不变.因此,进行随机投影转换可以起到压缩原始轨迹数据的作用.此外,选择较低的投影维数可以使得在同时获得投影矩阵和转换后数据的情况下也无法准确推测原始数据,能够扰动原始轨迹数据,起到隐私保护的作用[9].

若原始数据表示为 d 维向量 $\boldsymbol{X}$,利用 $d\times k$ 的随机投影矩阵 $\boldsymbol{R}$ 可以将 $\boldsymbol{X}$ 投影到 k 维空间:$X_{RP}=\boldsymbol{X}\cdot\boldsymbol{R}$,其中 $k<d$,随机矩阵 $\boldsymbol{R}$ 的选择有多种[8].为保护移动对象之间的距离,各个用户端在每个时间桶内的转换需要统一的随机投影矩阵,因此选择使用每个时间桶的初始时刻作为种子来生成随机投影矩阵. 为保证生成的矩阵一致,采用基于 GPS 的时间同步方法对所有用户端进行时间同步.转换后的数据为:(U,V,T),其中 $U=\boldsymbol{X}\cdot\boldsymbol{R}$,$V=\boldsymbol{Y}\cdot\boldsymbol{R}$,$T$ 取该时间桶的初始时刻.每个时间桶内轨迹数据转换的具体过程如算法 1 所示.

算法 1 轨迹数据转换算法.

输入:移动对象 o 的轨迹数据 tr;时间桶 T_{bin} 长度 d;投影维数 k.

输出:转换后的轨迹数据 P_{tr}

1) 对轨迹数据 tr 进行墨卡托投影转化

$x=\mathrm{mercator}(\mathrm{lat})$;$y=\mathrm{mercator}(\mathrm{lon})$.

2）划分时间桶，建立轨迹数据向量（$\boldsymbol{X}$，$\boldsymbol{Y}$，T）；

3）生成随机投影矩阵 $\boldsymbol{R}$ = generation(seed，d，k)；

4）对每个时间桶内的轨迹数据进行随机投影转换 $U = \boldsymbol{X} \cdot \boldsymbol{R}$；$V = \boldsymbol{Y} \cdot \boldsymbol{R}$；$P_{\text{tr}} = (U, V, T)$；

5）将转换后的轨迹数据 P_{tr} 发送给服务端.

2.2　相似轨迹挖掘

为了挖掘轨迹数据的相似模式，首先定义衡量轨迹数据间相似性程度的度量.

给定两个移动对象 o_1、o_2 在长度为 d 的时间桶内的轨迹数据，两者在该时间桶内轨迹的相似度为

$$\text{dist}(o_1, o_2) = \sqrt{\sum_{i=1}^{k} (x_i - x'_i)^2 + \sum_{i=1}^{k} (y_i - y'_i)^2} = \sqrt{\| X - X' \|_2^2 + \| Y - Y' \|_2^2}.$$

利用转换后的轨迹数据计算的相似度为

$$P_{\text{dist}}(o_1, o_2) = \sqrt{\sum_{i=1}^{k} (u_i - u'_i)^2 + \sum_{i=1}^{k} (v_i - v'_i)^2} = \sqrt{\| U - U' \|_2^2 + \| V - V' \|_2^2}.$$

由随机投影的性质[10]可知利用转换后的轨迹数据计算的相似度接近原始值，保持原始轨迹相似性.

挖掘相似轨迹的过程分为两部分：1）根据移动对象的轨迹数据密度进行初始聚类，定义为原子聚集；2）在给定时间范围内根据相似度（在某些时间桶内一直保持相似）寻找相似移动对象.选择支持 hamming 距离的 simhash 作为对应的 LSH 函数，将轨迹相似的移动对象映射到相同的哈希桶内.这样避免了传统方法中需要进行的交集运算，减小了计算复杂度.

定义 1（原子聚类）　在时间桶 T_{bin} 内，一个包含若干移动对象的集合 $O_{\text{atom}} = \{o_i \mid o_i \in O \wedge \text{dist}(o_i, o_{i'}) \leqslant r\}$ 称为原子聚类，其中：dist(o_i，$o_{i'}$) 为任意两移动对象在该时间桶内的相似度；r 为原子聚类相似性判定值.

定义 2（原子聚类索引）　原子聚类索引定义为二元组{cluID，Oset}，其中：cluID 为标识原子聚类的 ID；Oset 为原子聚类中包含的移动对象 ID 的集合.

定义 3（轨迹聚类索引）　轨迹聚类索引定义为{OID，cluset}，其中：OID 为标识移动对象的 ID；cluset 为移动对象在各个时间桶内的轨迹所在的聚类 ID 的集合.

相似轨迹挖掘过程如算法 2 所示.对于连续到来的每个时间桶内的轨迹数据 P_{tr}，算法首先根据密度进行初始原子聚类，为聚类建立原子聚类索引，然后根据阈值 r 动态维护原子聚类.在接下来的时间桶内追踪原子聚类，并以其为基础得到每个时间按桶内的聚类，将聚类规模大于阈值 σ 的聚类列为候选聚类.在最后时间桶内，利用 LSH 索引候选集中每个移动对象的轨迹聚类索引，根据 hamming 距离进行聚类得到相似轨迹集合.

算法 2　相似轨迹挖掘算法

输入：转换后的轨迹数据 P_{tr}，时间阈值 τ，相似轨迹集合的阈值 σ，原子聚类相似性判定值 r.

输出：相似轨迹集合.

1）采用基于密度的聚类方法进行初始的原子聚类得到初始原子聚类集合{clu}.

//更新原子聚类集合{clu}

2）原子聚类分裂.

For each $o_i \in$ clu

计算 o_i 与原子聚类中心 cen(clu) 的距离 dist(cen(clu)，o_i)；

If dist(cen(clu)，o_i) > r Then

将 o_i 作为一个新的原子聚类分裂出去；

将更新后的原子聚类添加到原子聚类集合{clu}；

3）原子聚类合并.

For clu_i，$\text{clu}_j \in$ {clu}

计算 clu_i、clu_j 的中心距离 dist(cen(clu_i)，cen(clu_j))；

If dist(cen(clu_i)，cen(clu_j)) $\leqslant r$ Then

将 clu_i 和 clu_j 合并，并将两者从{clu} 中删除，将新的合并聚类加入{clu}；

4）聚类.

For each 时间桶 T_{bin}

将各个原子聚类作为对象进行基于密度的聚类；if 聚类大小|clu| $\geqslant \sigma$ Then

将 clu 加入候选聚类集合{clu}；

5）相似轨迹聚类.

For each $o_i \in$ {clu}

对其轨迹聚类进行 LSH 索引；

分别取出每个哈希桶内的对象进行聚类（度量为 hamming 距离，阈值为 $|T_{\text{bins}}| - \tau$），将结果加入相似轨迹集合.

2.3　时间复杂性分析

算法 1 是在用户端运行，每个移动对象各自处理自己的轨迹数据，是并行的，因此分析单个移动对象的时间复杂度.假设轨迹数据长度为 d，墨卡托投影转换需 $O(d)$，随机投影转换需 $O(dk)$，

算法 1 的时间复杂度为 $O(d + dk)$.

算法 2 在服务端运行,是相似轨迹挖掘的执行方.对于包含 n 个移动对象的轨迹数据集,基于密度的聚类的时间复杂度为 $O(n^2)$.算法 2 包含初始阶段、聚类更新阶段和相似轨迹聚类阶段.初始聚类时间复杂度 $O(n^2)$.聚类更新的时间复杂度为 $O(n + m^2)$,m 为原子聚类的个数,其中:n 为分裂所需时间;m^2 为合并所需时间.基于 LSH 的移动对象的轨迹聚类时间复杂度是 $O(nn_h)$,其中 $n_h \ll n$, 是新的搜索空间的大小.由于轨迹数据都压缩为原始长度的 k/d,因此算法 2 所需时间复杂度为 $O(k(n^2 + m^2 + nn_h)/d)$.比较传统方法,一方面节省了交集计算所需的时间 $O(nl)$(l 为所有候选集的总计大小),另一方面由于数据压缩计算量都降为原来的 k/d.

3 实验与分析

实验通过测试本文方法 PSimTra 的效率和有效性来评估其性能.实验选取了与本文方法相近的 BU 和 SC 方法进行对比,需要指出的是 BU 和 SC 方法没有隐私保护功能.

实验分别选取真实数据集 Microsoft Geolife 与人工数据集进行测试.Geolife 数据集包含一系列带有时间戳的轨迹数据点,每个数据点包含纬度、经度和海拔高度信息.为了测试在不同参数设置下的算法性能,生成了两个人工数据集.实验中 DataSet_1 表示 Geolife 数据集,DataSet_2、DataSet_3 表示人工数据集.本实验环境为 Intel Core i7 处理器,8 GB 内存,1 TB 硬盘,64 位 Microsoft Windows 7 操作系统,使用 64 位的 Matlab(2012a)测试.

实验首先测试算法 1 将轨迹数据进行扰动压缩后对原始数据间的欧式距离的保护程度.分别测试原始轨迹数据间的距离和投影后数据间的距离,然后计算二者间的相对误差.如图 2 所示,相对误差随着 k/d 的增加而降低,因此选择合适的投影维数可以有效保护原始轨迹数据间的欧式距离.

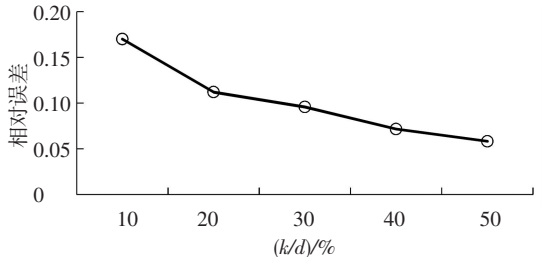


图 2 不同 k/d 值下的投影前后距离的相对误差

针对算法 2,实验首先测试了 3 种方法在不同数据集上的运行时间. 实验中利用随机投影将轨迹数据压缩为原来的 1/2(即 k/d = 1/2). 如图 3 所示,由于数据压缩以及避免了交集操作,PSimTra 在 3 个测试数据集上的运行时间要小于 BU 和 SC 算法.需要注意的是 y 轴为对数刻度,可以看出 PSimTra 在不同数量级下的表现.

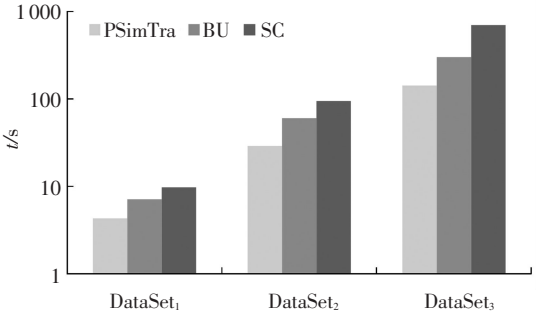


图 3 不同数据集上的执行时间比较

实验测试了相似轨迹数据集的阈值 σ 对时间效率的影响. 测试在数据集 DataSet_3 上进行,如图 4 所示,相似轨迹数据集的阈值越大,聚类阶段加入聚类集合的移动对象越少,需要进行 LSH 索引的对象越少,相似轨迹的搜索空间越小,相应的运行时间下降.此外,由于 BU 方法的过滤机制使得交集计算随 σ 而减少,运行时间也降低,因此也可见交集计算对算法运行时间的影响.

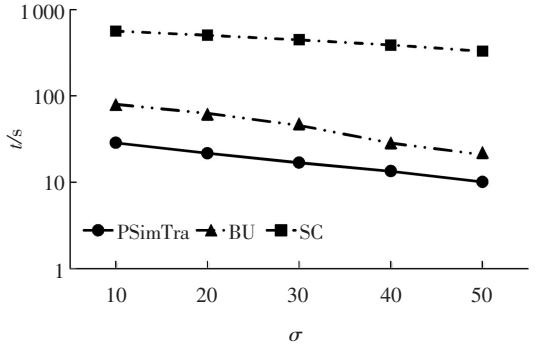


图 4 集合大小阈值对执行时间的影响

为了评估发现相似轨迹数据的质量,实验测试了不同长度的查询时间段下算法的准确率,准确率指结果集中实际的相似轨迹所占的比例.实验数据集 DataSet_2 上进行.如图 5 所示,PSimTra 的准确率随着时间阈值的增加而增加,达到一定数值时开始趋于稳定,PSimTra 最高可达的准确率略低于 BU、SC 方法,但是在时间阈值较小的情况下高于 BU、SC 方法.

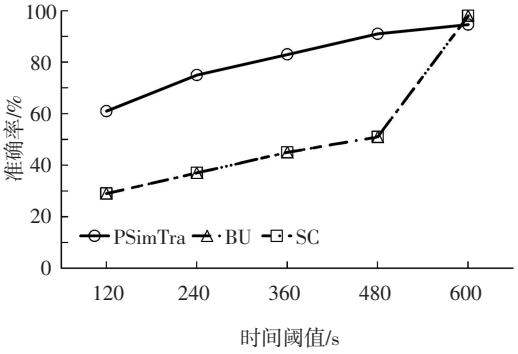


图 5　时间阈值对准确率的影响

4　结　论

1)提出了一种能够保护用户轨迹数据隐私的在线相似轨迹挖掘方法,该方法在用户端进行数据转换,利用随机投影技术压缩和扰动原始轨迹数据,将转换后的数据发送给服务端.服务端首先将接收到的数据实时聚类,然后在查询时间范围内,利用 LSH 技术快速寻找该段时间内的相似轨迹数据集,避免了传统方法中计算复杂度较高的交集计算,达到在线挖掘相似轨迹的目的.

2)实验结果表明,本文方法可以减少挖掘过程的时间提升效率,并能有效的发现相似轨迹.

参考文献

[1] 刘大有,陈慧灵,齐红,等. 时空数据挖掘研究进展[J]. 计算机研究与发展,2013,50(2): 225-239.

[2] JEUNG H, YIU M L, JENSEN C S. Trajectory pattern mining[C]//In Computing with Spatial Trajectories. New York: Springer, 2011:143-177.

[3] TANG Luan, ZHENG Yu, YUAN Jing, *et al*. On discovery of traveling companions from streaming trajectories[C]//Proceedings of the 28th International Conference on Data Engineering(ICDE). Washington, DC: IEEE Press, 2012: 186-197.

[4] QI Guojun, AGGARWAL C C, HUANG T S. Online community detection in social sensing[C]//Proceedings of the sixth ACM international conference on Web search and data mining. New York: ACM, 2013: 617-626.

[5] MUCKELL J, HWANG J H, LAWSON C T, *et al*. Algorithms for compressing GPS trajectory data: an empirical evaluation [C]//Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York: ACM Press, 2010:402-405.

[6] HAI P N, IENCO D, PONCELET P, *et al*. Mining representative movement patterns through compression [C]//Advances in Knowledge Discovery and Data Mining. Berlin: Springer, 2013: 314-326.

[7] 吴英杰,唐庆明,倪巍伟,等. 基于聚类杂交的隐私保护轨迹数据发布算法[J]. 计算机研究与发展,2013,50(3): 578-593.

[8] 霍峥,孟小峰. 轨迹隐私保护技术研究[J]. 计算机学报,2011,34(10): 1820-1830.

[9] LIU Kun, KARGUPTA H, RYAN J. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining[J]. IEEE Transactions on Knowledge and Data Engineering,2006,18(1): 92-106.

[10] ACHLIOPTAS D. Database-friendly random projections: Johnson-Lindenstrauss with binary coins[J]. Journal of Computer and System Sciences,2003,66(4): 671-687.

[11] BAWA M, CONDIE T, GANESAN P. LSH forest: self-tuning indexes for similarity search[C]//Proceedings of the 14th International Conference on World Wide Web. New York: ACM. Press, 2005: 651-660.

(编辑　张　红)