

带相对时间的轨迹序列模式挖掘

刘素杰

(中国矿业大学 计算机学院, 江苏 徐州 221116)

摘 要: 针对带有时间约束的序列模式挖掘算法时空效率低的问题, 引入相对时间, 提出带相对时间的轨迹序列模式挖掘. 该算法利用相对时间作为约束条件, 首先基于网格划分计算相对时间内的网格密度, 接着对密度网格进行扩展得到兴趣区域, 然后在兴趣区域的基础上挖掘轨迹序列模式. 使用真实数据进行实验, 实验表明, 与传统的序列模式挖掘算法相比, 该算法的挖掘效率在时间和空间上都有明显提高.

关键词: 相对时间; 网格密度; 兴趣区域; 序列模式

中图分类号: TP312

文献标识码: A

文章编号: 1000-7180(2013)11-0108-04

Trajectory Sequential Pattern Mining with the Relative Time

LIU Su-jie

(School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)

Abstract: For the problem that the low efficiency of time constrained sequential pattern mining algorithm in the aspect of time and space, this paper proposes trajectory sequential pattern mining with the relative time. The algorithm uses the relative time as a constraint condition. Firstly, it computes grid density of the relative time based on grid division, and then obtains regions of interest through extension dense grid. What is more, conduct trajectory sequential pattern mining on the basis of regions of interest. Experiments on real dataset show that this algorithm maintains better efficiency in time and space performance than traditional sequential pattern mining algorithms.

Key words: relative time; grid density; regions of interest; sequential pattern

1 引言

随着 GPS 功能设备和无线传感器技术的快速发展, 很方便获取海量的数据信息, 通过对这些信息的分析, 从中挖掘出重要的轨迹模式^[1]已成为数据挖掘的重要研究课题. 序列模式挖掘最早是由 Agrawal^[2]等人对超市中的购物数据分析后提出的, 通过找出在交易数据库中超过最小支持度的所有候选序列模式来发现客户的购买活动规律. 近年来, 序列模式挖掘的应用范围不只局限于交易数据库, 在 Web 访问新型应用数据源、疾病诊断、DNA 分析等许多领域已有针对性研究.

目前, 关于序列模式已有大量研究. 较典型的是

GSP^[2]算法和 PrefixSpan^[3]算法, 利用提前设定的约束条件和最小支持度, 对序列数据库反复扫描, 但缺少时间约束. 文献[4]利用基于 EM 的聚类算法研究序列模式, 但只考虑了空间位置. 文献[5]将项与项的时间范围设定为多个可变区间, 并结合剪枝技术发现模式, 但划分标准不明确, 挖掘结果不精确.

针对传统序列模式挖掘算法时空效率较低的问题, 本文引入相对时间, 提出带相对时间的轨迹序列模式挖掘 (Trajectory Sequential Pattern Mining with the Relative Time, TSPM_RT). TSPM_RT 基本框架如图 1 所示. TSPM_RT 分为两个阶段: 预处理阶段和模式发现阶段. 在第一阶段, 算法基于网格密度发现兴趣区域, 将兴趣区域转换为兴趣区域

收稿日期: 2013-01-08; 修回日期: 2013-02-25

基金项目: 教育部高等学校博士学科点专项科研基金项目 (20110095110010); 中央高校基本科研业务费专项资金 (2012DXS02)

序列,并计算兴趣度.在第二阶段,利用兴趣度查询相似用户,并通过序列投影扩展对序列删除或合并,挖掘出序列模式.实验表明,TSPM_RT的挖掘效率较传统序列模式算法有明显提高.

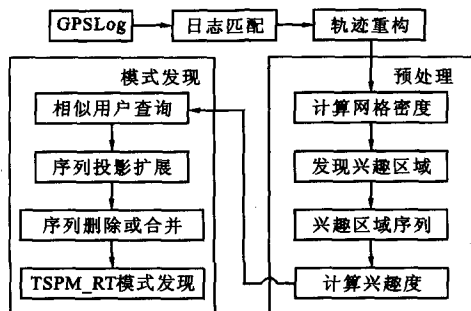


图1 TSPM_RT 基本框架

2 序列模式挖掘

序列模式是移动对象中大量序列的集合,反映移动对象在相似的时间间隔内访问相同位置的序列,包含两方面:(1)空间兴趣区域;(2)两个临近兴趣区域的时间间隔.给出两条序列模式例子如下:

车站 $\xrightarrow{20\text{min}}$ 广场 $\xrightarrow{3\text{h}30\text{min}}$ 歌剧院 (a)

车站 $\xrightarrow{12\text{min}}$ 大桥 $\xrightarrow{20\text{min}}$ 学校 (b)

从上面例子可知,序列模式(a)是典型的游客行为,表示游客从车站到歌剧院,该过程共需3h50min.模式(a)并未指定特定的旅行路线,只限定旅行时间.而序列模式(b)表示学生从车站步行到达学校,中间的大桥是必经之路,即路线确定,该过程共需32min.

两条序列模式的时间间隔基本代表模式中单个移动对象经过指定路径的时间.单个移动对象不要求同步,允许有独立的开始时间,只要满足在相似时间间隔内经过相同的位置即可.

对于给定的移动对象数据集 D ,最小支持度为 δ ,时间容忍度为 τ ,空间邻域函数为 $F: R^2 \rightarrow p(R^2)$,则挖掘结果为带有时间信息的序列模式 S ,且满足 S 的平均支持度不小于最小支持度 δ .

3 TSPM_RT 算法

对于普通的大型数据库,轨迹序列模式挖掘过程反复迭代,时空开销大,挖掘结果包含所有序列模式,没有一定的时间针对性,实用性不高.

鉴于上述问题,本文在挖掘前添加时间约束,利用时间约束可减少计算量,降低挖掘的时空复杂性,提高挖掘效率.

相对时间是参考给定时间得到的时间,通过将整体时间划分为许多小时时间段来提取.利用统计学原理中的期望和方差指导时间段的划分是划分的理论依据.

定义1 (兴趣区域 ROI) 设定最小支持度为 δ ,给定的网格 G 表示移动对象的运动平面,网格密度为 $G(i,j)$ ($1 \leq i \leq m$ 且 $1 \leq j \leq n$),网络单元格的平均密度为 σ ,网格 G 中的网络单元格组成集合 R ,则兴趣区域定义如下:(1) 每一个 $r \in R$ 组成的矩形区域;(2) 各矩形区域间没有交集;(3) 矩形区域中 σ 不小于 δ ;(4) 矩形区域是最大的.

网格密度是经过给定网格中轨迹的条数.网格密度计算前,首先划分时间段取相对时间.然后确定网格,对选定的相对时间内的数据集中每条轨迹执行网格遍历,当一条轨迹经过该网格则网格密度值增加1,直到所有轨迹被遍历完为止,并储存结果,即网格密度.

3.1 发现兴趣区域

兴趣区域是轨迹序列模式挖掘中的重要组成部分,通过扩展密度网格得到,它把空间点和区域坐标联系起来,将空间平面划分为一系列有意义的区域.

基于上述描述,给出算法1的伪代码过程,如算法1所示.

算法1: FindROI

输入:带密度 $G(i,j)$ 的网格 G ,最小支持度 δ .

输出:矩形区域集 R ,即兴趣区域 ROI.

```

1: Initialize  $R = \Phi$ ,  $(i,j) \in G$ ,  $G' = \{G(i,j) \geq \delta\}$ ;
2: foreach  $(i,j) \in G'$ , do  $f(i,j) = \text{false}$ ,  $G(i,j)$  order by DESC;
3: while  $f(i,j) = \text{true}$ , do  $r = \{(i,j)\}$ ;
4: foreach  $\text{dire} \in \{\text{up}, \text{down}, \text{left}, \text{right}\}$ ,
5: do  $r_{\text{dire}} = r$ , spread(dire);
6: if  $r_{\text{dire}} \in G$ ,  $\text{avg}(r_{\text{dire}}) \geq \delta$ , then  $d = \{\text{dire}\}$ ;
7: if  $d \neq \Phi$ ,  $\text{dire} \in d$ , then  $\text{dire} = \max(\text{avg}(r_{\text{dire}}))$ ,  $r = r_{\text{dire}}$ .
8: until  $d = \Phi$ ;
9: foreach  $(i,j) \in r$ , do  $f(i,j) = \text{true}$ ;
10:  $R = R \cup \{r\}$ ;
11: Return  $R$ ;
End
  
```

算法1利用迭代的方法发现兴趣区域 ROI. 步骤(1-3)找出网格密度不小于最小支持度的所有密度网格;(4-5)对每一个密度网格构建矩形区域,并在上、下、左、右四个方向扩展;(6)对扩展后的矩形区域在每个方向上给出判断,只留下网格密度不小于最小支持度的矩形区域,形成新的矩形区域;(7-

11)在新的矩形区域中,寻找平均密度最大方向上的矩形区域,重复上述步骤,直到将矩形区域扩展到最大,则矩形区域即为兴趣区域 ROI.

3.2 TSPM_RT 模式发现

利用算法 1 发现的兴趣区域将原始的大量轨迹压缩为轨迹数据的集合,然后利用算法 2 挖掘轨迹序列模式.

定义 2 (前缀 prefix) 假设序列中元素的项按照字典序排列. 对于给定的序列 $\alpha = \langle e_1, e_2, \dots, e_n \rangle$ 和序列 $\beta = \langle e'_1, e'_2, \dots, e'_m \rangle$ ($m \leq n$), 如果 $e_i = e'_i$ ($i \leq m-1$), $e_m \subseteq e'_m$, 且 $(e_m - e'_m)$ 中的项都在 e'_m 之后, 则称 β 是 α 的前缀 prefix. 而 $|\text{prefix}|$ 表示前缀的长度, 即前缀 prefix 中包含的项的个数.

定义 3 (投影 Projection) 给定序列 α 和序列 β , 假如 β 是 α 的子序列 (即 $\beta \subseteq \alpha$), 则 α 关于 β 的投影 α' 必须满足两个条件: (1) β 是 α' 的前缀; (2) α' 是 α 上满足上述条件的最大子序列.

定义 4 (后缀) 对于给定的序列 $\alpha = \langle e_1, e_2, \dots, e_n \rangle$ 和 $\beta = \langle e_1, e_2, \dots, e'_m \rangle$ ($m \leq n$), 如果 β 是 α 的前缀, 则 $\alpha'' = \langle e''_m, e_{m+1}, \dots, e_n \rangle$ 是序列 α 关于序列 β 的后缀, 且满足条件 $e''_m = (e_m - e'_m)$.

定义 5 (投影数据库) 假设 β 是序列数据库 S 的一个序列模式, 则 β 在 S 中的投影数据库是 S 中所有以 β 为前缀的序列相对于 β 的后缀, 即 $S|_{\beta}$.

基于上述描述, 给出算法 2 实现轨迹序列模式挖掘的伪代码过程, 如算法 2 所示.

算法 2: TSPM_RT

输入: 轨迹集 T_m , 网格 G_0 , 空间邻域函数半径 ξ , 最小支持度 δ , 时间支持度 τ .

输出: 轨迹序列模式 (S, A) .

1: Initialize $S = 0; T_0 = \{T_m \times \{\Phi\}\};$

2: if $T_s \neq \Phi$, then $T_{s+1} = \Phi;$

3: foreach $(T, \text{prefix}) \in T_s$, do

4: if $|\text{prefix}| \geq 3$, then GainRelativeTime(A)

5: Output(prefix, A);

6: $T = \text{Traj}(A);$

7: $G = \text{ComputeGridDensity}(T, G_0, \xi);$

8: $\text{ROI} = \text{FindROI}(G, \delta);$

9: $\text{ROIS} = \text{Translate}(\text{ROI}, T);$

10: foreach $r \in \text{ROI}$, do

11: if $\text{sup}(r) \geq \delta$, then

12: $\text{ROIS}' = \text{ProjectionSpread}(T, r, \text{ROIS});$

13: $T_{s+1} = T_{s+1} \cup \{(\text{ROIS}', A), (\text{prefix}, r)\};$

14: $S++;$

15: Return $S;$

End

算法 2 是基于投影数据库进行的. 步骤(1-4)对已存在的投影轨迹的前缀初始化, 并给出前缀的限制条件, 当前缀的长度不小于 3 时, 记录下相对时间 A ; (5-9) 将 A 返回, 用于投影, 把不符合 A 约束的轨迹删除, 重新获取符合要求的新轨迹 T , 利用 T 划分网格, 通过网格密度的计算和扩展获取兴趣区域 ROI, 并将其转换为兴趣区域序列 ROIS 来存储; (10-12) 对 ROIS 的每个序列, 若支持度不小于最小支持度, 就通过投影扩展序列的前缀; (13-15) 删除新投影数据库中不合要求的轨迹序列, 得到新的轨迹序列 ROIS', 将之前的序列与 ROIS' 合并, 即序列模式 (S, A) .

在网格、最小支持度和相对时间等约束条件的限制下, 算法 2 的时间复杂度是 $O(|G| \log |G|)$, 其中, G 表示网格密度. 利用算法 2, 节省大量挖掘时间, 降低了数据集的规模和轨迹数据研究的复杂性.

4 实验及分析

本文实验的数据集是大西洋飓风数据, 实验中共选择 82 条轨迹数据, 2338 个轨迹点, 取最小支持度的范围是 0.1%~0.85%, 空间邻域函数的半径范围是 5~15, 时间支持度的范围为 100~500, 网格的空间平面大小为 100×100 . 实验环境为: 操作系统为 Windows XP SP3, CPU 为 Intel(R) Core(TM)2 Duo, 3 GB 物理内存.

本文实验将 TSPM_RT 算法与经典的 PrefixSpan^[3] 算法进行多角度的比较. PrefixSpan 算法是序列模式挖掘领域被广泛认可的一种算法. PrefixSpan 算法利用数据库投影技术挖掘序列模式, 首先通过投影, 把序列数据库分割成许多很小的投影数据库, 挖掘频繁项, 将其作为下次投影的前缀序列模式. 然后递归地构建下一层的投影数据库继续挖掘, 直到挖掘出所有的序列模式. PrefixSpan 算法在序列模式挖掘时不产生候选序列, 可高效地处理大序列数据库.

4.1 轨迹条数与兴趣区域

实验一: 设最小支持度 $\delta = 0.25\%$, 网格宽度 size = 3, 空间邻域函数半径 $\xi = 9$, 时间支持度 $\tau = 500$, 网格的空间平面大小为 100×100 , 实验结果如图 2 所示. 图 2 表明, 随着轨迹数目的不断增加, 兴趣区域数目呈增长趋势, 且轨迹数目越多, 增长趋势越明显.

4.2 时间效率比较

实验二: 将飓风数据在 TSPM_RT 和 PrefixSpan 算法上进行序列模式挖掘. 实验表明, 两种算法发现的序列模式基本相同, 表明 TSPM_RT 算法切

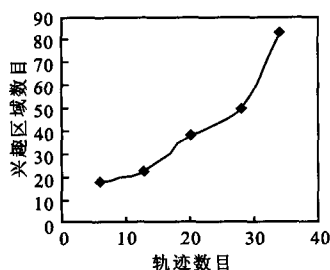


图2 兴趣区域数目与轨迹数目

实可行。

实验三:比较 TSPM_RT 算法和 PrefixSpan 算法的时间效率。实验结果如图 3 所示。图 3 表明,当最小支持度增加时,TSPM_RT 算法的运行时间明显少于 PrefixSpan 算法。因为 TSPM_RT 算法在序列模式挖掘之前利用相对时间作约束条件,只对兴趣区域内的数据集投影,投影数据库数少,运行时间少。而 PrefixSpan 算法通过递归构建大量投影数据库,并对投影数据库反复扫描,消耗大量时间。

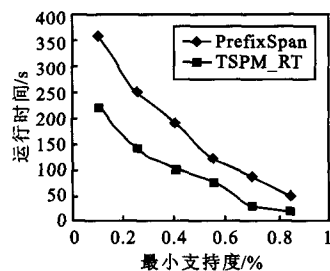


图3 运行时间与最小支持度

4.3 空间效率比较

实验四:比较 TSPM_RT 和 PrefixSpan 算法的内存消耗,即空间效率。实验结果如图 4 所示。图 4 表明,随着最小支持度的增加,TSPM_RT 算法比 PrefixSpan 算法内存消耗更少。这同样由于 TSPM_RT 算法采用相对时间进行约束后投影数据库比 PrefixSpan 算法更少,因此内存消耗更少。

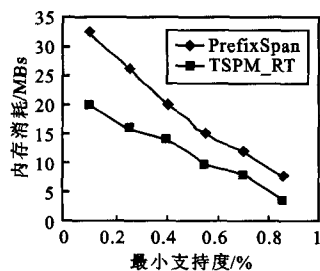


图4 内存消耗与最小支持度

由图 3 和图 4 可知,TSPM_RT 算法在时空性能上比 PrefixSpan 算法更优越。

本文还进行了其他实验,总结如下:(1)随着空间邻域函数半径的增大,运行时间和内存消耗增加;

(2)当时间支持度变化时,实验结果与经计算 TAS^[7]挖掘序列模式的结果类似;(3)当增加样本中的点数量时,运行时间呈近似线性变化。

5 结束语

针对传统轨迹序列模式挖掘结果多且时空效率低的问题,本文引入相对时间,提出带相对时间的轨迹序列模式挖掘 TSPM_RT。TSPM_RT 首先利用均匀网格对轨迹运动平面进行划分,并利用网格中经过的轨迹条数计算相对时间内的网格密度,然后扩展密度网格发现兴趣区域,将其转换为兴趣区域的序列,并在兴趣区域序列的基础上挖掘轨迹序列模式。实验表明,与传统的序列模式挖掘算法相比,该算法的挖掘效率在时间和空间性能上都有显著提高。

下一步工作将利用前期工作中相对时间内的轨迹序列模式预测移动对象未来时刻的位置,并进一步提高位置预测的效率和准确性。

参考文献:

- [1] Fosca Giannotti, Mirco Nanni, Dino Pedreschi, et al. Trajectory pattern mining [C]//Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: ACM, 2007: 330-339.
- [2] Srikant R, Agrawal R. Mining sequential patterns: generalizations and performance improvements [C]//Proc International Conference on Extending Database Technology. Avignon, France: IEEE, 1996.
- [3] Pei J, Han J. Prefixspan: mining sequential patterns efficiently by prefix-projected pattern growth [C]//Proceedings of the 17th International Conference on Data Engineering. Washington DC: IEEE Computer Society, 2001.
- [4] Gaffney S, Smyth P. Trajectory clustering with mixtures of regression models [C]//Proc 5th ACM SIGKDD Int'l Conf on Knowledge Discovery and Data Mining. San Diego, California: IEEE, 1999.
- [5] 李广原, 马楠, 曹丹阳. 一种多时间间隔序列模式挖掘算法 [J]. 微电子学与计算机, 2012, 29(4): 10-13.
- [6] Giannotti F, Nanni M, Pedreschi D. Efficient mining of sequences with temporal annotations [C]//Proc SIAM Conference on Data Mining. Bethesda, MD, USA: IEEE, 2006: 346-357.

作者简介:

刘素杰 女,(1987-),硕士研究生。研究方向为知识发现与数据挖掘。