TRAODGrid：基于 Grid 空间划分的高效离群轨迹检测方法

唐良 唐常杰 姜页希 李川 段磊 曾春秋 徐开阔
(四川大学计算机学院 数据库与知识工程研究所 四川省成都市 610065)
(tangliang@cs.scu.edu.cn, tangchangjie@cs.scu.edu.cn)

摘 要 为在海量离群轨迹数据的挖掘中提升算法运行效率，本文提出一种高效的离群轨迹挖掘算法 TRAODGrid。该算法通过挖掘离群轨迹点探测离群轨迹。本文提出的轨迹向量度量方法可以有效检测出轨迹点和轨迹分段在空间位置和轨迹方向上的离群性，并且通过 Grid 空间划分法，提高算法的运行效率。真实数据实验测试表明，TRAODGrid 算法比最新的 TRAOD 离群轨迹挖掘算法效率高出 2 个数量级。

TRAODGrid: An Efficient Trajectory Outlier Detection Algorithm with Grid-based Space Division

Tang Liang, Tang Changjie, Jiang Yexi, Li Chuan, Duan Lei, Zeng Chunqiu, Xu Kaikuo
(School of Computer Science and Technology, Institute of Database and Knowledge Engineering, Sichuan University, Chengdu Sichuan 610065)

Abstract Trajectory outlier detection is a widely used data mining application. This paper proposes a novel detection method called TRAODGrid to increase the efficiency of trajectory outlier detection. TRAODGrid detects the trajectory outlier based on finding trajectory point outliers. The concept of trajectory vector measurement can detect the positional and angular outliers in trajectory points and trajectories. By the space division, the efficiency of trajectory outlier detection can be improved significantly. Experiments show that, compared with the most recent algorithm TRAOD, our TRAODGrid surpasses it by 2 orders of magnitudes.

Key words Outlier analysis；Trajectory mining；Space division；

关键词 离群分析；轨迹挖掘；空间划分；

中图法分类号: TP301.6

1. 引言

离群点检测是数据挖掘的重要研究课题。离群点是指某些与数据集中数据在行为或模式上与众不同的数据对象。这些离群点数据对象本身可能隐藏重要的信息，或者可能是特别令人感兴趣的[1]。离群点挖掘具有广泛的应用，诸如信用卡欺诈检测，网络入侵判别等等。已有的离群点检测方法包括基于分布[2]，基于距离[3-6]，基于密度[7, 8]，基于偏差[9]等。

Knorr 等人提出方法通过提取轨迹的特征向量，利用常规离群检测方法来实现[3, 4, 5]。但特征向量内的属性都是轨迹的整体属性，因此轨迹的部分子段的异常可能被均化而不易被检测出。如图 1 中的 TR_3 轨迹中有部分异常，但是在上述整体特征属性无法体现。

J.-G Lee 等人提出 TRAOD 检查方法利用模式识别领域的 Hausdorff 距离来计算轨迹中两两子段之间距离 [11, 12] 。同时，TRAOD 中也引入了轨迹粗分(coarse-partition)来加快算法的效率。TRAOD 使用的 Hausdorff 距离度量是由任意两个有向线段之间的平行距离，垂直距离，夹角距离三部分加权求和求得[10, 11, 12]。Hausdorff 距离度量方法并不满足标准欧氏空间的准则，即距离的对称性与三角不等关系。任意 3 条轨迹分段 L_i, L_j, L_k，并不一定满足 $dist(L_i, L_j) = dist(L_j, L_i)$与 $dist(L_i, L_j)+dist(L_j, L_k) \geqslant dist(L_i, L_k)$，*dist* 为 Hausdorff 距离度量公式。Hausdorff 距离度量方法并不满足标准欧氏空间的准则，常规空间划分索引并不能在 Hausdorff 距离的轨迹分段数据空间内[10]。当轨迹数据集很大时，如果没有空间索引的支持，离群轨迹的挖掘效率十分低下。

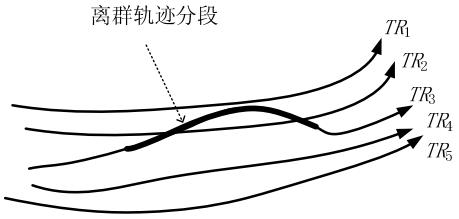

图 1 离群轨迹示例

本文的主要工作包括：(1)提出一种新的离群轨迹的判定方法。通过寻找每条轨迹上的离群轨迹点，计算其组成的离群轨迹分段的总长度来判定整条轨迹是否是离群轨迹。(2)提出一种新的高效离群轨迹检测方法 TRAODGrid。TRAODGrid 采用基于单元的空间划分方法。（3）真实数据测试结果表明，TRAODGrid 的运行效率高出 TRAOD 2 个数量级。

2. 预备知识

2.1 轨迹

定义1. 轨迹(Trajectory).在多维空间上随时间变化一有序的多维数据集合，称为轨迹，记为$TR_i= p_1p_2…p_{numi}$，其中$p_j(1 \leq j \leq num_i)$是一个d维的数据点，也称轨迹点。num_i为TR_i中数据点的个数。其中线段$L_k=p_ip_{i+1}…p_j(i<j\leq num_i)$是$TR_i$的一个轨迹分段*t-partition*。

符号约定. $S(TR_i)$表示轨迹TR_i中所有数据点构成的集合。$||X_i-X_j||$表示任意两个多维向量X_i,X_j的欧氏空间距离

收稿日期：
基金项目：国家自然科学基金（编号：60773169）；"十一五"国家科技支撑计划（编号：2006BAI05A01）

(Euclidean distance)。轨迹TR_i的长度表示为$len(TR_i)=\sum_{t=i}^{num_i-1}\| p_t - p_{t+1}\|$，轨迹分段$L_k= p_ip_{i+1}\ldots p_j$ ($i<j\le num_i$)的长度表示为$len(L_k)=\sum_{t=i}^{j-1}\| p_t - p_{t+1}\|$。

数据集中所有的轨迹集合记为集合$\boldsymbol{I}$ = {TR_1, TR_2,…, TR_{tnum}}, $tnum$为$\boldsymbol{I}$中轨迹的个数|$\boldsymbol{I}$|。离群轨迹集合记为集合$\boldsymbol{O}$={O_1,O_2,…,O_{onum}}, $onum$为$\boldsymbol{O}$中离群轨迹的个数|$\boldsymbol{O}$|。

2.2 轨迹点

观察1 图2中，轨迹数据集$\boldsymbol{I}$={TR_1 TR_2 TR_3 TR_4}，全部数据点组成的集合为$\boldsymbol{I_P}$。轨迹分段P1P2为一个轨迹TR_2中的某条轨迹分段。P1,Q1, Q2,…, P2在空间位置和切向量都与其它数据点有较大差异，在$\boldsymbol{I_P}$中是离群点。若考虑轨迹分段，由Hausdorff距离中指标则容易发现轨迹分段P1P2与其它轨迹分段差异较大，因此P1P2是离群轨迹分段。Hausdorff距离度量方法通过平行，垂直以及方向三个方面考虑有向线段之间的差异。在欧氏几何理论中，线段由点构成的点集，则任何有向线段可以看成是内部的点及其运动方向构成。所以，本文将通过分析轨迹分段内的数据点，判断轨迹分段是否离群。在轨迹数据集中，如果一条轨迹分段是离群的，则其内的若干数据采样点也应该是离群的，反之亦然。

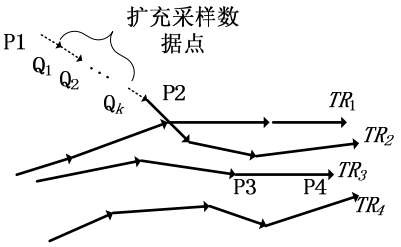

图2 数据采样点扩充

定义1表明在轨迹$TR_i = p_1p_2p_3\ldots p_j\ldots p_{len_i}$ 中，点p_j(1 ≤j≤num_i)都是运动物体的离散数据采样点。随着采样频率的提高，TR_i会逼近真实的运动轨迹。本文通过分析轨迹的数据点之间的性质来挖掘轨迹间的性质。下面定义中相关的符号定义列在表1。

定义2 轨迹向量（Trajectory Vector）. 一条轨迹TR_i中的一个数据点p_j ($j\ne num_i$)，num_i 为TR_i中数据点个数，p_j的轨迹向量表示为tv_j= (p_j, v_j)，p_j是数据点的空间向量，v_j是该点在TR_i上的单位切向量$v_j=(p_{j+1}-p_j)/\| p_{j+1}-p_j\|$。

如果TR_i内的数据点都是d维向量，则其数据点的轨迹向量tv_j是2d维向量。

轨迹向量间的距离度量(Trajectory Vector Distance). 给定任意两个2d维的轨迹向量tv_i= (p_i, v_i)与tv_j= (p_j, v_j)，其轨迹向量的距离定义为

$$d(tv_i,tv_j)=\sqrt{(\| p_i-p_j\|\cdot w_p)^2+(\| v_i-v_j\|\cdot w_v)^2}$$，其中w_p和w_v是用户给定的权值参数(w_p≥0, w_v≥0)，$\| X_1-X_2\|$表示向量X_1和X_2之间的欧氏空间距离。

权值w_p和w_v分别代表轨迹上一个采样点在空间位置差异和运动方向差异上的影响比重。容易证明，轨迹向量的距离公式满足度量空间(metric space)的性质，即空间上任取3个轨迹向量tv_i, tv_j, tv_k都满足:

$$d(tv_i, tv_j) = d(tv_j, tv_i),$$
$$d(tv_i, tv_k) \le d(tv_i, tv_j) + d(tv_j, tv_k)$$

邻近轨迹点(Neighboring Trajectory Point)**.**同一空间内的任意两个轨迹点p_i, p_j，当且仅当其满足: $d(tv(p_i), tv(p_j))$ ≤D，则称p_i和p_j互为邻近轨迹点，$tv(p_i)$与$tv(p_j)$互为邻近轨迹向量。D为用户给定的邻近距离参数。

定义3 离群轨迹点(Trajectory Point Outlier).一个数据点p($p\in TR_i$)为离群轨迹点，当且仅当其满足:

$$\sum_{TR_j\in I-\{TR_i\}}|CS(TR_j,p,D)|<pct\cdot\sum_{TR_k\in I}|S(TR_k)|,$$

其中D, pct为用户给定参数。$CS(TR_i, p_j, D)$是p_j在轨迹TR_i上的邻近轨迹点集，且 $p_j\notin S(TR_i)$ 。

2.3 离群轨迹分段与离群轨迹

离群轨迹分段(*t-partition* outlier).轨迹TR_i中的一条轨迹分段$L_k=p_ip_{i+1}\ldots p_j$($i<j\le num_i$, num_i为TR_i中数据点的个数)为离群轨迹分段，当且仅当其满足:

$$若j<num_i,\quad \bigcup_{t=i}^{j}\{p_t\}\subseteq OS(TR_i);$$
$$若j=num_i,\quad \bigcup_{t=i}^{j-1}\{p_t\}\subseteq OS(TR_i)。$$

定义4 离群轨迹(Trajectory outlier). 一条轨迹TR_i为离群轨迹，当且仅当其满足:

$$\sum_{L_k\in OTL(TR_i)}len(L_k)\ge F\cdot len(TR_i)$$

F为用户参数。

表 1 符号定义

符号	定义
$len(TR_i)$	轨迹 TR_i 的长度
$len(L_k)$	轨迹分段 L_k 的长度
$d(tv_i, tv_j)$	轨迹向量 tv_i 与 tv_j 之间的距离
$S(TR_i)$	TR_i 的所有数据点集合
$tv(p)$	数据点 p 的轨迹向量
$CS(TR_i, p_j, D)$	p_j 在轨迹 TR_i 上的邻近轨迹点集，即 {p | $p\in S(TR_i)$ ∧$d(tv(p), tv(p_j))\le D$ }
$OS(TR_i)$	轨迹 TR_i 中所有离群轨迹点集合
$OP(TR_i)$	轨迹 TR_i 内所有离群轨迹分段集合
$OTL(TR_i)$	$\bigcup_{t=1}^{num_i-1}\{p_tp_{t+1}\mid p_tp_{t+1}\in OP(TR_i)\}$

图3中有4条轨迹TR_1,TR_2,TR_3,TR_4。P1,P2是TR_2的轨迹点，P3,P4是TR_3的轨迹点。P1和P2无论在空间位置上，还是TR_2上的切向量，与其他轨迹点(如P3,P4)都有较大的差异，因此，P1和P2属于离群轨迹点，$OS(TR_2)$={P1,P2}，轨迹分段L_1 =P1P2即是离群轨迹分段，$OTL(TR_2) = OP(TR_2)$ = {L_1}。若L_1的长度占整个TR_2的比例大于参数F，则TR_2就是一个离群轨迹。

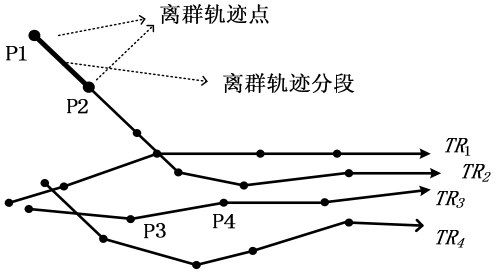

图3 离群轨迹示例

现实轨迹数据的采样规则可能不尽相同，部分轨迹跟踪的运动物体的运动速度也可能不同。如图4，P1P2判定为离群轨迹分段。由于P1P2轨迹分段较长，实际P1P2中的离群部分只有两端点附近少数部分，但在离群轨迹的判

断时是计算整条P1P2轨迹分段，因此造成一定误差。

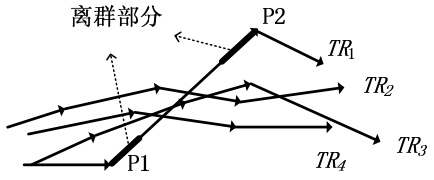

图4 不同长度的采样示例

通过线性插值方法，在轨迹分段每空间距离为u的位置上插入一轨迹点，整个插值过程的时间复杂度为$O(n_u)$，n_u为插入点的个数。虽然轨迹点的数量增大，但时间复杂度是线性的增长。此外，当采用本文4.2节介绍的Grid空间划分方法时，算法效率将进一步提高。

3. 离群轨迹检测算法

3.1 基本的离群轨迹检测算法

根据离群轨迹的定义，本文提出离群轨迹检测算法TRAODGrid。算法描述如下：

算法 1：TRAODGrid
输入：数据集 $\boldsymbol{I}$，参数 $pct, D, F, w_p, w_v, \{c^i\}$
输出：离群轨迹集 $\boldsymbol{O}$
步骤：

1. $\boldsymbol{O} = \varnothing$
2. MarkOutlierTrajPoints($pct, D, w_p, w_v, \{c^i\}$);
3. **for each** $TR_i \in \boldsymbol{I}$ **do**
4. $outlen = 0$
5. **for each** $L_k \in OTL(TR_i)$ **do**
6. $outlen = outlen + len(L_k)$
7. **if** $outlen \geq F \cdot len(TR_i)$ **then**
8. Insert TR_i to $\boldsymbol{O}$

说明: TRAODGrid 中第 3 行调用 MarkOutlierTrajPoints 子程序检测每个轨迹的离群轨迹点。本文后面部分会介绍具体实现。整个 TRAODGrid 的时间复杂度是 $O(n \cdot (|SI| + |SE| \cdot tvnum) + n)$，$n$ 为数据集中所有轨迹的轨迹点数量之和。

3.2 基于Grid空间划分的离群轨迹点检测算法

搜索每一轨迹向量在距离范围 D 内的邻近点个数时，为了避免全局空间的搜索，本文用 grid 划分搜索空间成为若干等大的单元(cell)。每个单元在第 i 维上的长度为 c^i，i=1,2,…$2d$, $2d$ 是轨迹向量的维数。

轨迹向量间距离范围. 记轨迹向量 tv 在 Grid 中单元的坐标向量为 $G(tv)=(g^1,g^2,\ldots, g^{2d})$，其中 g^i (i=1,2,…$2d$) 为大于或等于 0 的整数。

性质 1 给定任意两个属于同一 Grid 空间的轨迹向量 tv_i 与 tv_j，已知 $G(tv_i)=(g^1{}_i, g^2{}_{i,\ldots,} g^{2d}{}_i)$，$G(tv_j)=(g^1{}_j, g^2{}_{j,\ldots,} g^{2d}{}_j)$，设向量 $S=(s^1,s^2,\ldots,s^{2d})=G(tv_j)-G(tv_i)$，函数 gd(s) = max(|s|-1, 0)，则有如下两个不等式成立:

$$d(tv_i, tv_j) \geqslant \sqrt{w_p{}^2 \sum_{k=1}^{d} (\mathrm{gd}(s^k)c^k)^2 + w_v{}^2 \sum_{k=d+1}^{2d} (\mathrm{gd}(s^k)c^k)^2} \quad (1)$$

$$d(tv_i, tv_j) \leqslant \sqrt{w_p{}^2 \sum_{k=1}^{d} ((\mathrm{gd}(s^k)+1)c^k)^2 + w_v{}^2 \sum_{k=d+1}^{2d} ((\mathrm{gd}(s^k)+1)c^k)^2} \quad (2)$$

证明 通过欧氏几何中的三角不等式可以直接证明。由于篇幅所限，具体证明忽略。

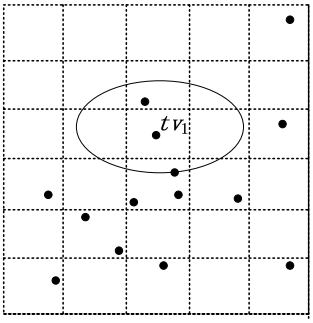

图 5 Grid 搜索空间划分

将邻近距离 $d(tv_i, tv_j)=D$ 代入不等式(1), 求出整数向量 S 的取值范围 SI。将 $d(tv_i, tv_j)=D$ 代入不等式(2)左边，求出整数向量 S 的取值范围 SO。设 Grid 中任一轨迹向量 tv 的单元坐标向量 $G(tv)$取值范围为集合 I_SET。集合 $SE=I\_SET-(SO \cup SI)$。

算法 MarkOutlierTrajPoints 首先计算出 $SE=\{SE_1,SE_2,\ldots,SE_N\}$, 其中 N 为 SE 内的元素个数。然后寻找任意一轨迹向量 p 的邻近轨迹向量时，只需要实际遍历单元坐标为 $G(tv(p))+SE_1$, $G(tv(p))+SE_2$, …, $G(tv(p))+SE_N$ 共 N 个单元内的轨迹向量。其它单元的单元坐标只有两种情况，若属于 SI，则其内的轨迹向量全部邻近 p; 若属于 SO，则其内的轨迹向量全部远离(不邻近)p。

算法 2 是本文提出的在基于 Grid 划分轨迹向量空间的方法上的离群轨迹点检测。其中，$grid[G]$表示 Grid 中单元坐标向量为 G 的单元，且其内的轨迹向量个数为 $grid[G].num$.

算法 2：MarkOutlierTrajPoints
输入：数据集 $\boldsymbol{I}$，参数 $pct, D, w_p, w_v, \{c^i\}$
输出：离群轨迹点
步骤：

1. $threshold = pct \cdot \sum_{TR_k \in I} |S(TR_k)|$
2. $grid$ = new Grid(D, $\{c^i\}$)
3. **for each** $TR_i \in \boldsymbol{I}$ **do**
4. **for each** $p \in S(TR_i)$ **do**
5. Insert $tv(p)$ to grid
6. Calculate SI,SO,SE
7. **for each** $cell$ in $grid$ **do**
8. **for each** tv_j in $cell$ **do**
9. $neigh = 0$
10. **for each** $S_i \in SI$ **do**
11. $neigh = neigh + grid[G(p)+S_i].num$
12. **for each** $S_i \in SE$ **do**
13. Check the trajectory vectors in $grid[G(p)+S_i]$, add the number of neighboring points to $neigh$
14. **if** $neigh > threshold$ **then**
15. mark tv_j as an outlier

说明: 算法 2 的时间复杂度为 $O(n \cdot (|SI| + |SE| \cdot tvnum))$, n 为数据集中所有轨迹的轨迹点数量之和, $tvnum$ 为 Grid 中每个单元内的平均轨迹向量数。$|SI|$，$|SE|$与数据集大小无关。随着 Grid 划分空间越细，$tvnum$ 越小。算法 2 的空间复杂度为 $O(\prod_{i=1}^{2d} \lceil dmax_i / c_i \rceil)$, $dmax_i$ 为轨迹向量空间中每维上的最大值。Grid 空间划分的方法在高维空间的时间复杂度很大，本文研究的轨迹向量空间维度为 4。

4. 实验

本文使用的实验数据集来源于1920年至2006年大西洋飓风中心运动轨迹记录数据集[1]，与文献[11,12]的数据集相同。整个数据集分成两块：1920 年到 1969 年全部共 590 条飓风数据，19217 个数据点; 1970 年到 2006 年选出 795 条飓风数据，共 24405 个数据点。

实验环境: Windows Server 2003 SP2 操作系统，Intel Pentium Dual 1.80G CPU, 2GB 内存。开发环境: Microsoft Visual Studio 2008。.NET Framework 2.0

图 6 和图 7 分别是针对 1970 年到 2006 年的 795 条轨迹数据集合和 1920 年到 1969 年的 590 条飓风数据集用本文算法 TRAODGrid 进行离群轨迹的检测。深色的轨迹表示 TRAODGrid 算法检测出的离群的飓风轨迹。算法的各项参数设定列在表 2。

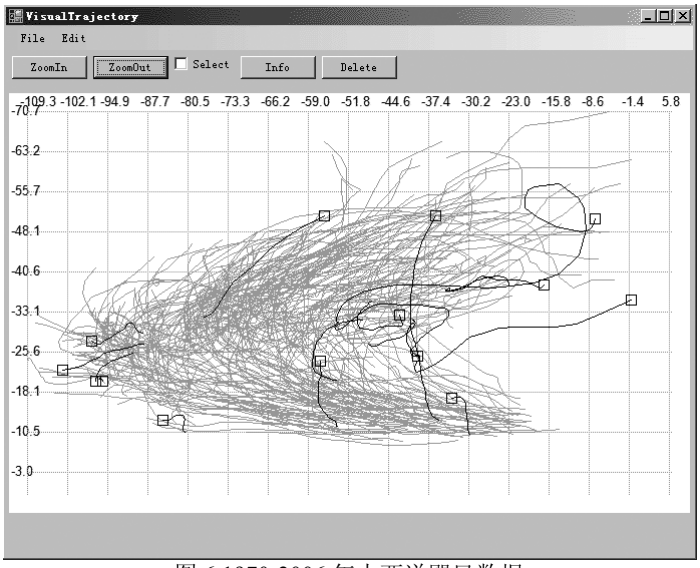

图 6 1970-2006 年大西洋飓风数据

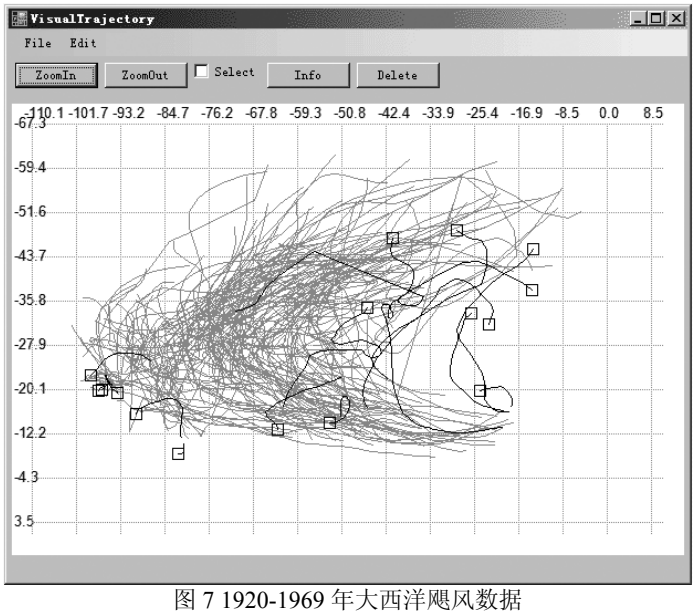

图 7 1920-1969 年大西洋飓风数据

实验过程中发现，数据集不变的情况下，不同比例的离群轨迹，算法的执行时间也是不同的。本文设定挖掘结果系数 *outlier_ratio* 为离群轨迹数量与整体数据集轨迹数量的比值 *outlier_ratio* = $|\boldsymbol{O}| / |\boldsymbol{I}|$。本文的离群轨迹挖掘效率对比实验中，不但保证了 TRAODGrid 和 TRAOD 的挖掘轨迹数据集 $\boldsymbol{I}$ 的相同，也保证了 *outlier_ratio* 相同。

表 2 TRAODGrid 参数设定

D	F	P	u	$\{w_p, w_v\}$	$\{c^1,c^2,c^3\}$
30	0.5	0.02	1.0	{1.0,3.0}	{7.0,7.0,0.3}

表 3 TRAOD 参数设定

Parameter name	Candidate set	Parameter name	Candidate set
D	{90, 85, 80}	w_θ	{0.1, 0.2, 1}
$w_\perp$	{0.1, 0.2, 1}	p	{0.95,0.97,0.98}
$w_\parallel$	{0.1, 0.2, 1}	F	{0.2,0.25,0.3}

表 4 和 5 是 TRAODGrid 与 TRAOD 在 1970-2006 年和 1920-1969 年大西洋飓风数据集上的对比。其中 *outlier_ratio*≈5%, Trajectory Set size 是挖掘数据集中轨迹的条数。其他两列分别是 TRAODGrid 与 TRAOD 算法的执行时间，单位为千分之一秒，其中除去了轨迹数据从磁盘装载的 I/O 读取时间。TRAODGrid 的参数列在表 2。实验过程中发现 TRAOD 中各参数的变化对离群轨迹的挖掘结果有很大的影响。为了保证 TRAOD 的 *outlier_ratio* ≈5%，实验中使用了表 3 中的候选参数集，通过不断组合各参数取值的搭配，直到选出 *outlier_ratio*≈5%的结果。

表 4 1970-2006 年数据集上的效率对比

Trajectory Set size	TRAODGrid execution time	TRAOD execution time
100	94	6720
200	234	23831
300	469	50051
400	782	93363
500	1235	151969

表 5 1920-1969 年数据集上的效率对比

Trajectory Set size	TRAODGrid execution time	TRAOD execution time
100	94	5984
200	328	32041
300	703	68922
400	1141	121344
500	1672	189.656

对比表 4 和表 5 可以看出，TRAODGrid 的执行效率几乎都高于 TRAOD 的 2 个数量级左右。TRAOD 虽然有 *corase-partition* 的方法减少近邻轨迹段的遍历个数[10]，但是与空间索引划分的方法比起来，算法时间复杂度依然很大。同时，TRAOD 采用的 Hausdorff 距离度量，在实际计算的中间计算步骤也比较繁多[10,11, 12]。

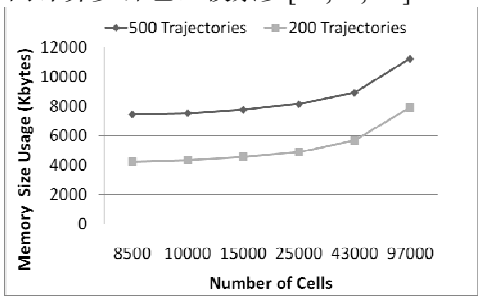

图 8 内存使用情况

本文分别使用了 500 条和 200 条 1970-2006 年的飓风数据来测量 TRAODGrid 实验程序运行时的最大进程内存占用情况。从图 8 中可以看到，随着 Grid 划分的空间单元数量增多，其内存占用呈线性增大。包括 Windows 进程自身所需内存，500 条轨迹的挖掘进程内存占用最大不超过 12MB。因此，本文中提出的针对离群轨迹挖掘所建立 Grid 空间划分方法的空间复杂度是可以接受的。

[1] http://weather.unisys.com/hurricane/atlantic/

5. 结语及未来工作

本文的主要工作包括: (1)提出一种新的离群轨迹的判定方法。通过寻找每条轨迹上的离群轨迹点，计算其组成的离群轨迹分段的总长度来判定整条轨迹是否是离群轨迹。(2)提出一种新的高效离群轨迹检测方法TRAODGrid。TRAODGrid 建立在基于单元的空间划分方法基础上。（3）在真实数据测试结果表明，TRAODGrid 的运行效率高于 TRAOD 大约 2 个数量级。

目前 TRAODGrid 算法和 J.-G Lee 等提出的 TRAOD 算法[10]对参数敏感度都较大，用户需要多次参数尝试，才能得到满意的结果。在今后的工作中，我们将在 TRAODGrid 基础上研究自适应变换参数的算法。

参 考 文 献

[1] J. Han and M. Kamber, Data Mining: Concepts and Techniques, 2nd ed. Morgan Kaufmann, 2006

[2] V. Barnett and T. Lewis, Outliers in Statistical Data. John Wiley & Sons, 1994.

[3] E. M. Knorr and R. T. Ng, “Algorithms for mining distance-based outliers in large datasets,” in Proc. 24th VLDB, New York City, New York, Aug. 1998, pp. 392–403.

[4] E. M. Knorr and R. T. Ng, “Finding intensional knowledge of distance-based outliers,” in Proc. 25th VLDB, Edinburgh, Scotland, Sept. 1999, pp. 211–222.

[5] E. M. Knorr, R. T. Ng, and V. Tucakov, “Distance-based outliers: Algorithms and applications,” VLDB Journal, vol. 8, no. 3, pp. 237–253, Feb. 2000.

[6] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient algorithms for mining outliers from large data sets,” in Proc. 2000 ACM SIGMOD, Dallas, Texas, May 2000, pp. 427–438.

[7] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: Identifying density-based local outliers,” in Proc. 2000 ACM SIGMOD, Dallas, Texas, May 2000, pp. 93–104.

[8] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, “LOCI: Fast outlier detection using the local correlation integral,” in Proc. 19th ICDE, Bangalore, India, Mar. 2003, pp. 315–326.

[9] C. C. Aggarwal and P. S. Yu, “Outlier detection for high dimensional data,” in Proc. 2001 ACM SIGMOD, Santa Barbara, California, May 2001, pp. 37–46.

[10] J.-G Lee, J. Han and X. Li, “Trajectory Outlier Detection: A Partition-and-Detect Framework,” in Proc. 2008 ICDE.

[11] J.-G. Lee, J. Han, and K.-Y. Whang, “Trajectory clustering: A partition-and-group framework,” in Proc. 2007 ACM SIGMOD, Beijing, China, June 2007, pp. 593–604.

[12] J. Chen, M. K. H. Leung, and Y. Gao, “Noisy logo recognition using line segment hausdorff distance,” Pattern Recognition, vol. 36, no. 4, pp. 943–955, Apr. 2003

唐 良 男， 1983年生，四川大学计算机学院，硕士研究生，主要研究方向为数据挖掘，数据库。

唐常杰 男，1946年生，教授，博导，主要研究方向为数据库系统，数据挖掘，知识工程等；中国计算机学会数据库专业委员会副主任，中国计算机学会人工智能与模式识别专业委员会委员。

姜页希 男，1985年生，硕士研究生，主要研究方向为数据挖掘，基因表达式编程等。

李 川 男，1977年生，博士，讲师，主要研究方向为数据挖掘，基因表达式变成等；中国计算机学会会员。

段磊 男，1982年生，博士研究生，主要研究方向为数据挖掘，基因表达式编程，进化计算等。

曾春秋 男，1983年生，硕士研究生，主要研究方向为数据挖掘。

徐开阔 男，1983年生，博士研究生，主要研究方向为数据库，数据挖掘，进化计算等。