ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 32,No. 2,Mar. 2017,pp. 382－389
DOI:10.16337/j.1004-9037.2017.02.020

http://sjcj.nuaa.edu.cn
E-mail:sjcj@nuaa.edu.cn
Tel/Fax:＋86-025-84892742

基于演化计算的异常轨迹并行检测算法

唐梦梦　吉根林　赵　斌

(南京师范大学计算机科学与技术学院,南京,210023)

摘　要:异常轨迹检测是轨迹数据挖掘研究领域的一个重要研究内容,基于演化计算的异常轨迹检测算法(Top-k evolving trajectory outlier detection, TOP-EYE)是一种有效的异常轨迹检测算法。不同于其他算法采用的轨迹距离计算方法,TOP-EYE 算法从轨迹的方向和密度角度出发,采用演化计算的方式检测异常。为了提高 TOP-EYE 算法对海量轨迹数据集异常检测的效率,本文在其基础上提出了基于 MapReduce 的异常轨迹检测并行算法(Parallel detecting abnormal trajectory based on TOP-EYE, PDAT-TOP),利用 MapReduce 并行计算的优势提高了异常轨迹检测的效率。将算法 PDAT-TOP 在 Hadoop 平台上加以实现,实验结果表明,算法 PDAT-TOP 能够有效地检测异常轨迹,并且具有较高的可扩展性和加速比。

关键词:异常轨迹检测;演化计算;并行异常轨迹检测;时空轨迹挖掘

中图分类号:TP39　　**文献标志码**:A

Parallel Algorithm for Detecting Trajectory Outliers Based on Evolutionary Computation

Tang Mengmeng, Ji Genlin, Zhao Bin

(School of Computer Science and Technology, Nanjing Normal University, Nanjing, 210023, China)

Abstract: Trajectory outlier detection is significantly important in the field of trajectory data mining. Algorithm TOP-EYE (Top-k evolving trajectory outlier detection) is an efficient algorithm for detecting abnormal trajectory. From the point of view of the direction and density, algorithm TOP-EYE takes use of the method of evolutionary computation to detect anomalies, which is different from other algorithms. To improve the efficiency of mining trajectory outliers from massive trajectory datasets, the parallel algorithm for detecting trajectory outliers based on evolutionary computation, called PDAT-TOP (Parallel detecting abnormal trajectory based on TOP-EYE), is proposed. The algorithm takes advantages of parallel computation to improve the efficiency of detecting abnormal trajectory. Algorithm PDAT-TOP is implemented on Hadoop. Experimental results demonstrate that the algorithm can effectively detect abnormal trajectory, and it has high scalability and better speedup.

Key words: trajectory outlier detection; evolutionary computation; parallel detecting abnormal trajectory; spatial temporal data mining

---

基金项目:国家自然科学基金(41471371)资助项目。
收稿日期:2015-03-10;**修订日期**:2015-08-30

引　言

随着 GPS 定位、RFID 定位以及基站定位等移动对象定位技术的发展,人们能够以较高的时空分辨率记录移动对象的位置历史数据[1,2]。然而,高时空分辨率轨迹数据的急剧增长,对轨迹数据的分析和挖掘提出了新的挑战。近年来,轨迹数据挖掘研究已成为数据挖掘研究领域的热点,其中包括:轨迹的聚类[3,4]、伴随模式挖掘[5,6]、频繁模式挖掘[7,8]以及异常轨迹检测[9]等。异常轨迹检测是指从轨迹数据集合中找出严重偏离正常模式的对象[10,11],它是轨迹数据挖掘领域的一个重要分支,被广泛应用于出租车欺诈、飓风路径变化和动物迁徙等异常行为识别领域。由于轨迹是由一系列相互关联的轨迹点组成,这使得传统的异常点检测算法无法直接用来检测异常轨迹。目前针对轨迹数据的异常轨迹检测算法比较少。2000 年,Knorr 等[12]提出将轨迹转换成多个独立属性组成的对象,利用基于距离的传统离群点检测算法检测异常轨迹;2007 年,Li 等[13]从人工智能的角度出发,提出一种基于分类器的异常轨迹检测算法,该算法通过将轨迹转换成具有代表性的移动特征,利用抽取的移动特征训练一个分类模型;2008 年,Lee 等[14]提出了异常轨迹检测算法(TRAjectory outlier dectection, TRAOD)算法,该算法根据最小描述长度原则[15]对轨迹进行粗分割和细分割,根据轨迹段的相似度,综合考虑异常轨迹段的比例和异常轨迹段的密度来确定异常轨迹;2010 年,Ge 等[16]采用演化方式检测异常轨迹,提出基于演化计算的异常轨迹检测算法(TOP-k evolving trajectory outlier detection, TOP-EYE)方法,该方法以演化计算方式计算轨迹在每个网格中的方向偏离和密度偏离积分,从而快速识别异常轨迹。但是 TOP-EYE 算法对于海量轨迹数据的运行效率较低,为此本文采用 MapReduce[17]并行计算模型将 TOP-EYE 算法并行化,提出了基于演化计算的异常轨迹并行检测算法(Parallel detecting abnormal trajectory based on TOP-EYE, PDAT-TOP),从而提高了异常轨迹检测效率。

1　背景知识

在 TOP-EYE 算法中,首先将连续空间分割成等大小的网格,然后建立概率模型将每个网格中轨迹的方向信息转化成一个八维的向量,用来记录该网格中轨迹在这八个方向的移动概率。下面给出文献[16]中定义的概率向量、方向异常指数、密度异常指数以及演化异常指数。

定义 1(时空轨迹)　时空轨迹是移动对象在空间上连续运动的位置信息在时间上的表示。$TDB=\{TR_1,TR_2,\cdots,TR_i,\cdots,TR_s\}$表示轨迹数据库,$TR_i=\{TraID,\ p_1,\ p_2,\cdots,p_j,\cdots,p_l\}$是一条轨迹,其中 $TraID$ 为轨迹标识,$p_j=\{x_j,\ y_j,\ t_j\}$表示移动对象 i 在 t_j 时刻的位于$\langle x_j,\ y_j\rangle$处。

定义 2(概率向量)　将地理空间分割成等大小的一系列网格,$G=\{\boldsymbol{g}_1,\boldsymbol{g}_2,\cdots,\boldsymbol{g}_i,\cdots,\boldsymbol{g}_m\}$表示网格集合,$\boldsymbol{g}_i=\langle p_1,p_2,p_3,p_4,p_5,p_6,p_7,p_8\rangle$表示一个向量,$p_j(1\leqslant j\leqslant 8)$表示网格 i 中轨迹在方向 j 上的概率,所以 $\boldsymbol{g}_i$ 也称为概率向量。

如图 1 所示,每个网格中有 8 个方向,并用 1～8 这 8 个数字对方向进行标识。每两个相邻方向之间的角度为 $\pi/4$,将网格中的所有轨迹方向简化成这 8 个方向。计算网格中所有轨迹方向在这 8 个方向上的概率,得到概率向量。

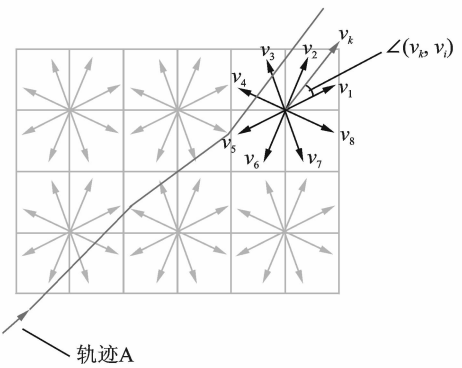

图 1　异常轨迹检测示例

Fig. 1　Illustrative example of detecting abnormal trajectory

定义 3(方向异常指数)[16]　设一条轨迹 TR_i 经过一系列网格,记作 $GL=\{\boldsymbol{g}_1,\boldsymbol{g}_2,\cdots,\boldsymbol{g}_n\}$。在这些网格中,若 TR_i 在方向上偏离大部分轨迹的方向,则该轨迹是方向异

常轨迹。可由式(1)计算轨迹 TR_i 在每个网格中的方向异常指数。

$$\mathrm{OScoreDir}=1-\sum_{k=1}^{K}qk\sum_{i=1}^{8}pi\cdot\cos\angle(\boldsymbol{v}_k,\boldsymbol{v}_i) \tag{1}$$

式中：K 表示轨迹 TR_i 在该网格中有 K 个方向，并用方向向量$\langle q_1,\cdots,q_K\rangle$表示 TR_i 的方向，q_k($1\leqslant k\leqslant K$)等于 $1/K$，$\cos\angle(\boldsymbol{v}_k,\boldsymbol{v}_i)$为向量 $\boldsymbol{v}_k$ 和 $\boldsymbol{v}_i$ 夹角的余弦值，如图 1 所示。

定义 4(密度异常指数)[16] 若轨迹 TR_i 经过的网格中，轨迹数量小于一定阈值，则该轨迹为密度异常。可由式(2)计算轨迹 TR_i 在每个网格中的密度异常指数。

$$\mathrm{OScoreDen}=\begin{cases}s & \mathrm{Density}<\tau\\ 0 & \text{其他}\end{cases} \tag{2}$$

定义 5(演化异常指数)[16] 演化异常指数既要考虑历史异常又要考虑实时异常，其中历史异常的影响应随着时间的推移逐渐衰减，所以异常指数以一种积累的方式随时间持续更新。演化异常指数的计算公式为

$$S_{ti}^{\Sigma}=S_{ti}+S_{ti-1}*\exp(-\lambda\Delta t_{i-1})+S_{ti-2}*\exp(-\lambda\Delta_{ti-2})+\cdots+S_{t0}*\exp(-\lambda\Delta t_0) \tag{3}$$

式中：S_{t0} 为式(1)和式(2)的和，即方向异常指数和密度异常指数的和；Δt_{i-k} 为 t_i 时刻和 t_{i-k} 时刻之间的时间差。当演化异常指数大于用户给定的阈值时，该轨迹可判定为异常轨迹。

2 TOP-EYE 算法

TOP-EYE 算法是一种基于演化计算的异常轨迹检测算法，主要考虑了两种类型的异常，即方向异常和密度异常。具体步骤分为以下 3 步：(1)将连续的空间离散成同等大小的网格；(2)利用概率模型将每个网格中的轨迹方向信息转换成一个八维向量，以表示轨迹在该网格中八个方向上的移动概率，即在一个固定区域通过总结大量轨迹在一段时间内的移动方向来产生方向趋势；(3)一旦某个被检测对象跨越该区域时，通过计算被检测对象的方向与总体方向的相似性值来实时地检测方向异常。此外，在该离散网格空间，可以方便地通过计算经过该网格的轨迹数量来判断被检测对象是否密度异常。

TOP-EYE 算法的检测流程如图 2 所示，利用轨迹数据库 TDB 为网格建立概率向量模型；对于待检测轨迹，可由式(2)和(3)计算实时异常指数 Oscore_k 和演化异常指数 Escore_k。若演化异常指数大于用户给定的阈值 E，则轨迹 TR_i 为异常轨迹。

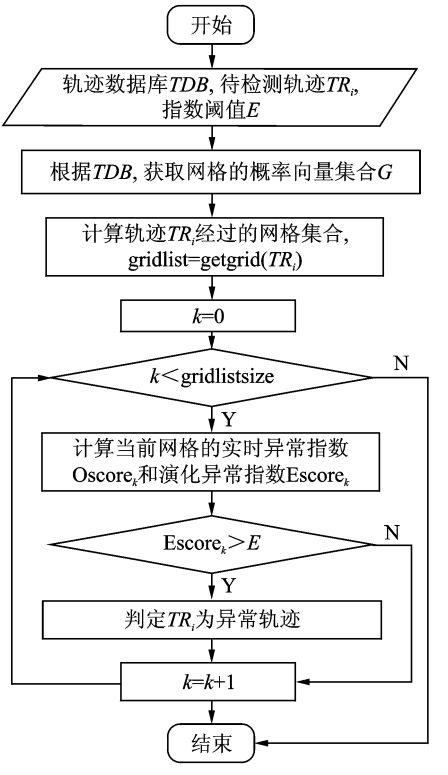

图 2 TOP-EYE 算法流程图

Fig. 2 Flow chart of algorithm TOP-EYE

3 PDAT-TOP 算法

随着轨迹数据库的增大，为了提高 TOP-EYE 算法的效率，本文基于 MapReduce 编程模型，提出基于演化计算的异常轨迹并行检测算法 PDAT-TOP，其总体框架如图 3 所示。它分为以下 3 个主要步骤：(1)地图网格化，根据地理范围，将地图划分成等大小的网格；(2)利用已知轨迹数据库，获得相应网格的轨迹密度和方向概率向量；(3)计算待检测轨迹经过的网格集合，查询网格的密度和方向概率，判断轨迹是否异常。

3.1 获取概率向量

获取网格信息的目的是用来检测轨迹,当轨迹经过一些网格时,若偏离网格中大部分轨迹的方向或者该网格中的轨迹数量较少,则该轨迹可能为异常轨迹。获取网格信息这一过程需要MapReduce的 Job 来完成,如图 3 总体框架的上半部分所示,具体步骤可见算法 1。

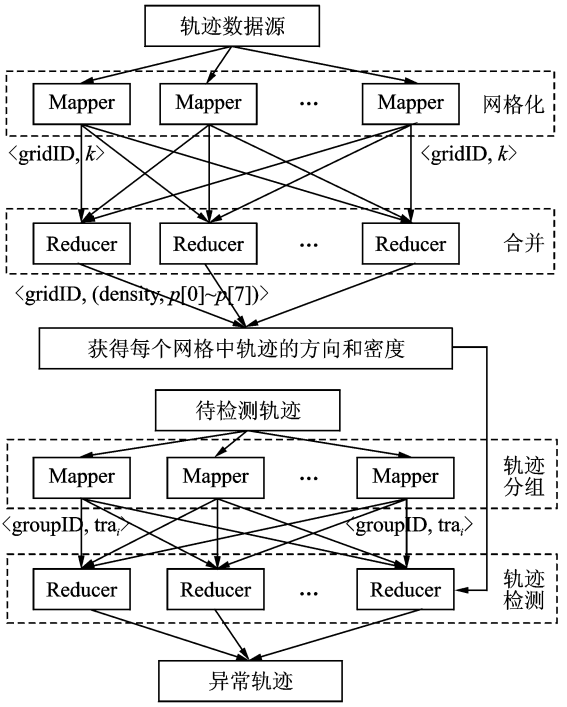

图 3 PDAT-TOP 算法总体框架

Fig. 3 Framework of PDAT-TOP algorithm

算法 1 获取概率向量

输入:轨迹数据库 TDB,密度阈值 τ

输出:每个网格的概率向量

map(key, value){//key 是偏移量,value 是轨迹 TR_i

1. for (j=1;j<len_i;j++)
2. subtra$_j$=TR_i[j,j+1]; //由相邻两个采样点组成轨迹段 subtra$_j$
3. **vector**$_j$=getv(subtra$_j$); //计算轨迹段 subtra$_i$ 的方向向量 **vector**$_j$
4. gridlist=getgridID(subtra$_j$); //获取轨迹段经过的网格,得到网格标识符集合 gridlist
5. k=getclosedir(**vector**$_j$); //在网格的 8 个方向,获取中与向量 **vector**$_j$ 夹角最小的方向标识 k
6. for each gridID in gridlist
7. output(gridID, k);
8. endfor
9. endfor

}

map 函数的输入数据为轨迹数据库,步骤 4 是将轨迹段对应到相应网格,获得网格列表;步骤 5 计算网格 8 个方向中与轨迹方向最近的方向,并代替原来的轨迹方向;最后,map 函数输出键值对〈gridID, k〉,用于计算网格的方向概率。

reduce (key, value){//key 为网格的标识符 gridID,value 为方向标识 k_i

1. density=getdensity (value, (); //计算网格的密度异常指数
2. sum=0; //统计网格中的方向向量总数量
3. for each k_i in value
4. sum++;
5. count[k_i]++; //统计网格中每个方向上的方向向量数量
6. endfor
7. for(j=0; j<8; j++)
8. p[j]=count[k_i]/sum; //计算网格每个方向的概率
9. endfor

10. output(key, (density, $p[0]\sim p[7]$)); //输出网格信息

}

reduce 函数输入的 key 值为网格标识 gridID，value 为方向标识 k_i，属于同一网格的方向标识被发送到同一个 reduce，用于建立该网格的方向概率模型。步骤 1 通过式(2)计算网格密度异常指数；步骤 8 计算网格中相应方向上的概率，概率值越大，该方向上的轨迹越多，反之，则轨迹越少；步骤 10 输出网格的方向和密度信息，用于下一步的异常轨迹检测。

3.2 检测轨迹

已知网格方向和密度信息，输入待检测轨迹，通过计算异常指数并与阈值进行比较，可以判断轨迹是否异常。这一过程同样需要一个 MapReduce 的 Job 来完成，具体框架如图 3 下半部分所示，具体步骤见算法 2。

算法 2 检测轨迹

输入：待检测轨迹 Tra_i，网格信息 gridinfor，阈值 E

输出：异常指数

map(key, value){// key 是偏移量，value 是轨迹 Tra_i

1. groupID=getID(key); //按照分组策略，计算小组标识 groupID
2. Output(groupID, Tra_i);

}

map 函数的功能是将待检测轨迹进行分组，并分发到不同的 reduce 函数，实现并行检测异常轨迹。因此，map 函数的输出键值对为〈groupID, Tra_i〉，其中 groupID 为小组标识。

reduce(key,value){//key 是小组标识 groupID，value 为属于同一组的待检测轨迹

1. for each Tra_i in value
2. gridlist=getgrid(Tra_i); //获得轨迹 Tra_i 经过的网格列表以及在网格中的方向向量
3. for(i=0; i<gridlist.size; i++)
4. $Oscore_i$=getOscore(gridlist.get(i), gridinfor); //根据网格信息，计算轨迹 Tra_i 在每个网格中的实时异常指数
5. Oscores.add($Oscore_i$);
6. endfor
7. $Escore_0$=Oscores.get(0);
8. for(i=1;i<Oscores.size;i++)
9. $Escore_i$=getEscore(Oscores.get(i),$Escore_{i-1}$); //计算轨迹的演化异常指数
10. if($Escore_i$>E)
11. Output("outlier"); //Tra_i 为异常轨迹
12. endif
13. endfor
14. endfor

}

检测异常轨迹的过程可通过运行一个 Job 实现，通过 map 函数对轨迹进行分组，输出〈groupID, Tra_i〉，由不同的 reduce 函数进行轨迹检测。在算法 2 的步骤 4 中，已知网格信息，根据式(1)和式(2)计算经过相应网格时的实时异常；在步骤 9 中，根据式(3)计算演化异常指数，若该指数值大于用户给定的阈值，则该轨迹为异常轨迹。

4 实验及结果分析

将 PDAT-TOP 算法在云计算平台上加以实现。本文的云计算实验环境共有 21 台服务器(1 个主节点,20 个从节点),每个节点配置相同。节点的处理器为 Intel(R) Xeon(R) CPU E5620 ,主频为 2.40 GHz,操作系统为 64 位 Debian Linux 操作系统,Hadoop 版本为 hadoop-0.20.2。实验采用的数据集为 2012 年 11 月 30 日北京约 12 000 辆出租车 24 h 内采集到的数据[18],总大小为 1.82 GB,平均每辆出租车共有 2 702 个采样点。从该数据集中分别截取了前 1 500 辆、3 000 辆、6 000 辆和 12 000 辆出租车的轨迹用于实验测试,其数据大小分别为 95 MB,185 MB,371 MB 和 741 MB。

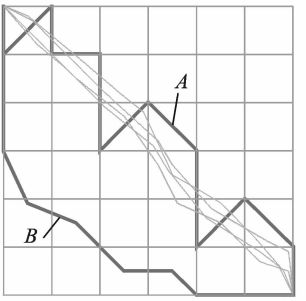

图 4 PDAT-TOP 算法检测结果示例

Fig. 4 Illustrative example of detecting abnormal trajectory for PDAT-TOP

图 4 给出了 PDAT-TOP 算法检测结果示例,已知用细线表示轨迹数据,经过轨迹检测,轨迹 A 和轨迹 B 为异常轨迹。从图 4 可发现,轨迹 A 的移动方向和细线表示的轨迹明显不同,所以方向异常指数值较大;轨迹 B 经过的网格中,大部分网格的密度为 0,这就导致密度异常指数值较大,所以轨迹 A 和轨迹 B 为异常轨迹。

图 5 为轨迹 A 和 B 的异常指数分布。从图 5(a)可以发现,由于轨迹 A 的方向变化明显,所以随着方向的不同,实时异常指数会存在变化,演化异常指数也会有相应变化并且值较大;图 5(b)为轨迹 B 的异常指数值,由于轨迹 B 经过的大部分网格中没有其他轨迹,所以轨迹 B 的实时异常指数达到最大值,演化异常指数值则随着时间递增。用户给定合适的阈值,则可判定轨迹 A 和轨迹 B 为异常轨迹。

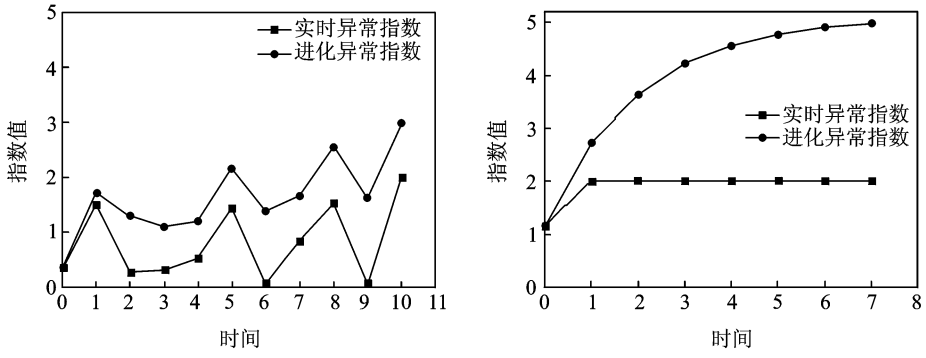

图 5 轨迹 A 和 B 的异常指数值

Fig. 5 Abnormal score of trajectories A and B

图 6 为 TOP-EYE 算法和 PDAT-TOP 算法在不同数据集下的运行时间,其中 PDAT-TOP 的 Hadoop 平台的计算节点数为 20 个。从图 6 可看出,TOP-EYE 算法的运行时间明显高于 PDAT-TOP;随着数据量的增加,TOP-EYE 算法运行时间增加的越多,而 PDAT-TOP 算法的时间则变化不大,这是由于 PDAT-TOP 算法充分利用了 MapReduce 编程模型的特点,从而提高了算法效率。

如图 7 所示,利用 741 MB 的轨迹数据集,测试 PDAT-TOP 算法在不同计算节点下的运行时间。

从图 7 可以发现，随着计算节点的增加，算法的运行时间逐渐减小，这也说明该算法具有较好的加速比；当计算节点数超过 10 时，时间减小的不是特别明显，这是由于当计算节点增多到一定程度时，通信时间将会占用主要时间，导致总运行时间减少的幅度变小。

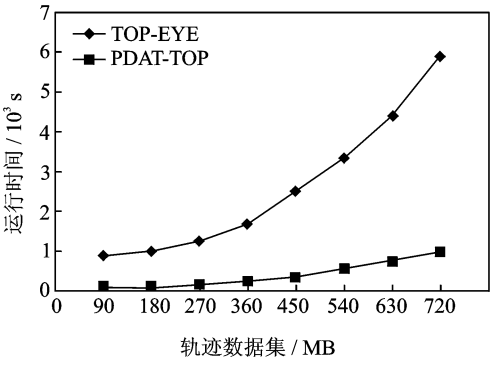

图 6 不同数据集下两种算法的运行时间

Fig. 6 Execution time of two kinds of algorithms on different datasets

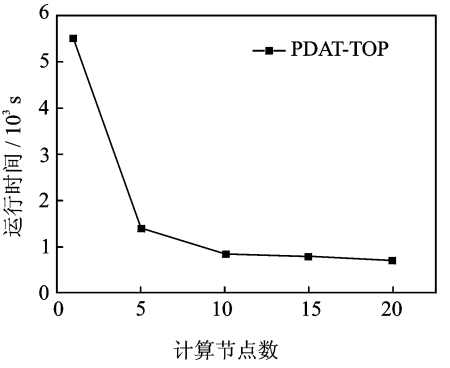

图 7 不同计算节点下的算法运行时间

Fig. 7 Execution time of algorithm PDAT-TOP

5 结束语

异常轨迹检测通常应用于物流运输、动物迁徙和交通管理等领域，随着移动对象定位技术的发展，积累的轨迹数据越来越多。面向海量轨迹数据集时，目前已有的一些算法难以在效率上满足人们的需求。本文在 TOP-EYE 算法的基础上提出了 PDAT-TOP 算法，该算法同时考虑了轨迹的方向和密度信息，并且通过地图网格化减少了计算量；但随着轨迹数据的增多，TOP-EYE 算法的效率越来越低。在此基础上，PDAT-TOP 算法利用了 Hadoop 并行计算平台的特点，让集群中的多个计算节点同时检测不同的轨迹，从而提高了算法效率并用于大规模异常轨迹检测。实验结果表明，随着计算节点的增加，本文提出的算法具有良好的可扩展性和加速比。

参考文献：

[1] Gupta M, Gao J, Charu A, et al. Outlier detection for temporal data: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 25: 1-20.

[2] 吉根林，赵斌. 时空轨迹大数据模式挖掘研究进展[J]. 数据采集与处理，2015，30(1)：47-58.
Ji Genlin, Zhao Bin. Research progress in pattern mining for big spatio-temporal trajectories[J]. Journal of Data Acquisition and Processing, 2015, 30(1): 47-58.

[3] Lee J G, Han J, Whang K Y. Trajectory clustering: A partition-and-group framework[C]// Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data. Beijing, China: ACM, 2007: 593-604.

[4] Zhang Z, Huang K, Tan T. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes[C]// Proceedings of the 18th International Conference on Pattern Recognition. Hong Kong, China: IEEE Computer Society Press, 2006, 3: 1135-1138.

[5] Uehara K, Seki K, Jinno R. Parallel distributed trajectory pattern mining using MapReduce[C]// Proceedings of the 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom). Taipei, China: IEEE Computer Society Press, 2012: 269-273.

[6] Giannotti F, Nanni M, Pinelli F, et al. Trajectory pattern mining[C]// Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2007: 330-339.

[7] Tang L A, Zheng Y, Yuan J, et al. On discovery of traveling companions from streaming trajectories[C]// Data Engineering

(ICDE), 2012 IEEE 28th International Conference on. Washington DC, USA: IEEE Computer Society Press, 2012:186-197.

[8] Aung H H, Tan K L. Discovery of evolving convoys[C]// 22nd International Conference on Scientific and Statistical Database Management. Heidelberg, Germany: Springer, 2010:196-213.

[9] Chen H, Yang C C. Intelligence and security informatics, techniques and applications[M]. Heidelberg: Springer, 2008:357-381.

[10] 姜金凤. 移动对象轨道异常检测算法的研究[D]. 南京:南京航空航天大学,2010:1-19.
Jiang Jinfeng. Research on moving object trajectories outlier detection algorithms [D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2010:1-19.

[11] 刘良旭,乔少杰,刘宾,等. 基于 R-Tree 的高效异常轨迹检测算法[J]. 软件学报,2009,20(9):2426-2435.
Liu Liangxu, Qiao Shaojie, Liu Bin, et al. Efficient trajectory outlier detection algorithm based on R-Tree[J]. Journal of Software, 2009,20(9):2426-2435.

[12] Knorr E M, Ng R T, Tucakov V. Distance-based outliers: Algorithms and applications [J]. The VLDB Journal—the International Journal on Very Large Data Bases, 2000,8(3/4):237-253.

[13] Li X, Han J, Kim S, et al. ROAM: Rule-and motif-based anomaly detection in massive moving object data sets[C]// Proceedings of the SIAM International Conference on Data Mining. Minneapolis, Minnesota: SIAM, 2007,7:273-284.

[14] Lee J G, Han J, Li X. Trajectory outlier detection: A partition-and-detect framework[C]// Proceedings of the 24th International Conference on Data Engineering. Washington: IEEE Computer Society, 2008:140-149.

[15] Grünwald P, Myung I J, Pitt M. Advances in minimum description length: Theory and applications[M]. Cambridge: MIT Press, 2005.

[16] Ge Y, Xiong H, Zhou Z, et al. Top-eye: Top-*k* evolving trajectory outlier detection[C]// Proceedings of the 19th ACM international Conference on Information and Knowledge Management. Toronto, Canada: ACM Press, 2010:1733-1736.

[17] Dean J, Ghemawat S. MapReduce: A flexible data processing tool[J]. Communications of the ACM, 2010,53(1):72-77.

[18] 数据堂. 某北方城市 12000 辆出租车 GPS 位置数据(2012 年 11 月)[EB/OL]. http://www.datatang.com/data/44502, 2013-09-16.
Datatang. Taxi GPS data of one city in North of China (201211) [EB/OL]. http://www.datatang.com/data/44502, 2013-09-16.

作者简介:

唐梦梦(1990-),女,硕士研究生,研究方向:时空轨迹挖掘,E-mail:dreamtang1016@163.com。

吉根林(1964-),男,教授,博士生导师,研究方向:数据挖掘技术与应用,E-mail:glji@njnu.edu.cn。

赵斌(1978-),男,博士,副教授,研究方向:数据挖掘技术与应用,E-mail:zhaobin@njnu.edu.cn。