doi:10.3969/j.issn.1001-4616.2019.01.010

基于聚类的兴趣区域间异常轨迹并行检测算法

许　振,吉根林,唐梦梦

(南京师范大学计算机科学与技术学院,江苏 南京 210023)

[摘要]　轨迹异常检测能够用来分析移动对象的异常运动行为,在交通运输、医疗监护等领域都有广泛应用.兴趣区域是移动对象集中活动的区域. 本文提出了一种新的兴趣区域间异常轨迹检测算法(Detecting Anomalous Trajectories Between Interest Regions,DATIR). 不同于已有的从局部采样点进行检测的算法,DATIR 算法综合考虑了轨迹的局部特征和全局特征,利用聚类方法检测兴趣区域间的异常轨迹,并能挖掘出兴趣区域间的正常路径. 为了提高海量轨迹数据的异常检测效率,在 DATIR 算法的基础上,提出了一种并行检测算法(Parallel Algorithm for Detecting Anomalous Trajectories Between Interest Regions,PDATIR). 实验结果表明,DATIR 算法能够有效地检测兴趣区域间的异常轨迹,并且能够检测出兴趣区域间的正常轨迹;PDATIR 算法在大数据集上表现出了明显的性能优势,具有较好的可扩展性和较高的加速比.

[关键词]　异常轨迹检测,兴趣区域,聚类,MapReduce
[中图分类号]TP39　[文献标志码]A　[文章编号]1001-4616(2019)01-0059-06

An Algorithm for Detecting Anomalous Trajectory Between Interest Regions Based on Clustering

Xu Zhen,Ji Genlin,Tang Mengmeng

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

Abstract:Trajectory anomaly detection can be used to analyze the anomalous behavior mobile objects,and it has been widely used in transportation,medical monitoring and other fields. Interest region is an active activity area. They can be railway stations,airports,schools,shopping malls and so on. A new algorithm for detecting trajectory outliers between interest regions,called DATIR(Detecting Anomalous Trajectories between Interest Regions),is proposed. Unlike the existing algorithm to detect from the local sample points,algorithm DATIR takes into account the local and global characteristics of the trajectory,and uses clustering method to detect the normal path and anomalous trajectory between interest regions. Algorithm DATIR has a wide range of application areas,such as taxi fraud monitoring,road planning,and so on. In order to improve the efficiency of mining trajectory from massive trajectory datasets,the parallel algorithm for detecting trajectory outliers based on MapReduce framework,which is called PDATIR(Parallel algorithm for Detecting Anomalous Trajectories between Interest Regions),is presented. The experimental results demonstrate that algorithm DATIR can effectively detect the anomalous trajectories between regions of interest,and can mine the normal path. Algorithm PDATIR shows obvious performance advantages over large data sets,and it has the high scalability and good speedup ratio.

Key words:trajectory outlier detection,region of interest,clustering,MapReduce

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. 通过分析这些数据,人们能够了解到移动对象的移动特点,同时能够给道路等公共基础设施的建设提供决策. 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、伴随模式挖掘、频繁模式挖掘以及异常轨迹检测等[1-4]. 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[5-7],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

收稿日期:2018-06-16.
基金项目:国家自然科学基金(41471371).
通讯联系人:吉根林,博士,教授,博士生导师,研究方向:数据挖掘与大数据分析技术. E-mail:glji@njnu.edu.cn

诈、飓风路径变化、动物迁徙等异常行为识别领域[8-9].

两个兴趣区域间的异常轨迹检测在交通分析中有着至关重要的作用,例如,某辆出租车在两个兴趣区域间的行驶轨迹与大部分出租车的行驶轨迹不同,那么该出租车可能存在欺骗乘客的行为;另一方面,当两个兴趣区域间某条道路的移动对象数量过多时,可能出现交通拥堵,而异常的轨迹可提供另一种道路选择.

目前,基于两个特定区域之间异常轨迹的发现已有相关的研究. 2011 年,Daqing Zhang 等人提出了 iBAT 算法,该算法将轨迹转换成网格序列,并利用隔离森林(Isolation Forest)的方法来检测异常轨迹[10];2011 年,Yong Ge 等人设计了一个出租车欺诈检测系统,该系统通过分析行驶路线和行驶距离两个方面来检测出租车行驶欺诈[11];2013 年,Fontes 等人提出了另一种检测兴趣区域间异常轨迹的算法,该算法首先获取标准的行驶路径,并在此基础上检测出异常轨迹[12]. 上述几种算法都是通过检测局部采样点的方法来检测异常轨迹,只考虑了轨迹的局部特征,这就会使得对于噪声数据过于敏感,并且不能发现兴趣区域间有代表性的标准路径.

本文提出了一种新的用于检测兴趣区域间异常轨迹的算法(detecting anomalous trajectories between interest regions,DATIR). 该算法从轨迹本身出发,使用动态时间规整距离(dynamic time warping,DTW)作为轨迹相似性的度量方法[13-14],综合了轨迹的局部特征和整体特征两方面;并利用 DBSCAN 算法将兴趣区域间的轨迹进行聚类,每个簇代表一条标准路径,簇中的轨迹为正常轨迹,而不在簇中的轨迹为异常轨迹. 为了提高检测效率,本文还利用 Hadoop 平台并行地检测不同兴趣区域间的异常轨迹,提出了并行检测算法(parallel algorithm for detecting anomalous trajectories between interest regions,PDATIR).

1　相关知识

定义 1(轨迹)　轨迹是由一系列有序且多维的点组成,即 $TR_i=\{p_1,p_2,\cdots,p_i,\cdots,p_n\}$,其中 $p_i=(x_i,y_i,t_i)$,$t_1<t_2<\cdots<t_i<\cdots<t_n$.

定义 2(轨迹距离)　若 TR_1、TR_2 是两个兴趣区域间的两条轨迹,$DTW(TR_i,TR_j)$ 表示轨迹 TR_i、TR_j 之间的动态时间规整距离,变量 m 和 n 分别表示 TR_i、TR_j 中采样点的个数,p_1^i 和 p_1^j 分别是轨迹 TR_i、TR_j 中的第一个采样点,$\mathrm{dist}(p_i,q_i)$ 是两个点间的欧式距离,$Rest(TR_i)$、$Rest(TR_j)$ 表示轨迹 TR_i、TR_j 去掉第一个采样点后,其余点组成的轨迹. 这两条轨迹间的动态时间规整距离如公式(1)所示.

$$DTW(TR_i,TR_j)=\begin{cases}0, & \text{if } m=n=0,\\ \infty, & \text{if } m=0 \text{ or } n=0,\\ \mathrm{dist}(p_1^i,p_1^j)+\min\begin{cases}DTW(Rest(TR_i),Rest(TR_j))\\ DTW(Rest(TR_i),TR_j)\\ DTW(TR_i,Rest(TR_j))\end{cases}, & \text{otherwise.}\end{cases}\tag{1}$$

定义 3(正常轨迹)　若 $C_1,\cdots,C_k$ 是两个兴趣区域间所有轨迹聚类形成的簇,轨迹 TR_i 是经过这两个兴趣区域的一条轨迹,若 TR_i 在 $C_1,\cdots,C_k$ 的任意一个簇内,那么 TR_i 为正常轨迹.

定义 4(异常轨迹)　若 $C_1,\cdots,C_k$ 是两个兴趣区域间所有轨迹聚类形成的簇,轨迹 TR_i 是经过这两个兴趣区域的一条轨迹,若 TR_i 不在 $C_1,\cdots,C_k$ 的任何一个簇内,那么 TR_i 为异常轨迹,记作 O_i.

如图 1 所示,R_1 和 R_2 是两个兴趣区域,它们可以是商场、公园、车站、学校等人群活动较为密集的区域,$TR_1,TR_2,TR_3,\cdots,TR_9$ 是在这两个兴趣区域间活动的移动对象的轨迹. 从图 1 可以看出,TR_2,TR_3,TR_4,TR_5 经过的路径和 TR_6,TR_7,TR_8 经过的路径是比较统一且正常的,而 TR_1 和 TR_9 都偏离了大部分对象走的路径,故为异常轨迹. 利用

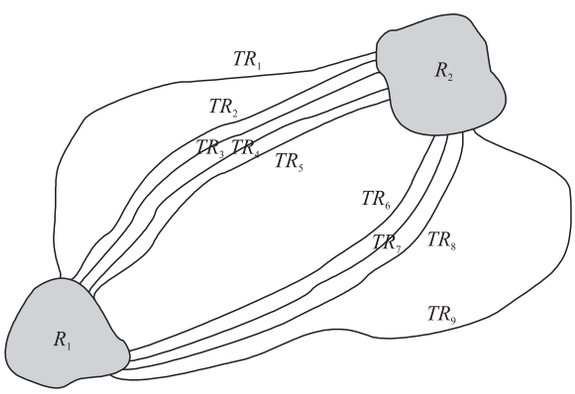

图 1　异常轨迹示例

Fig. 1　An illustrative example of anomalous trajectory

聚类算法,可以将 R_1 和 R_2 之间的轨迹聚成两类,也就代表了可选择的两条正常路径,不属于这两条路径的轨迹即可判定为异常轨迹.

2　异常轨迹检测算法

异常轨迹检测算法 DATIR 的基本思想是通过聚类算法发现兴趣区域间的异常轨迹,主要包括以下 4 个步骤:

(1)兴趣区域建立

(2)轨迹划分

(3)轨迹聚类

(4)判断轨迹是否异常

首先输入轨迹数据和兴趣区域,其中兴趣区域的大小和形式取决于不同的应用场景,它可以是地区、密集区域、热点、重要场所等等;接下来,根据轨迹的起点和终点所在的兴趣区域,将轨迹划分到对应的兴趣区域间;对于任意两个兴趣区域间的所有轨迹,采用 DTW 距离进行轨迹相似性度量,并将它们聚类;最后,判断轨迹是否异常,在两个兴趣区域间,若只存在一个簇,则表示这两个兴趣区域间只有一条标准路径,不在簇中的轨迹则为异常轨迹;若存在多个簇,则表示这两个兴趣区域间有多条路径可选择,不在簇中的轨迹同样为异常轨迹;若不存在簇,则表示这两个兴趣区域间无标准路径,也就不存在异常轨迹. 具体步骤见算法 1.

算法 1　异常轨迹检测算法

输入:轨迹数据集 TD、兴趣区域集合 Rlist

输出:异常轨迹

datir(*TD*,*Rlist*){//*TD* 表示轨迹数据集,*Rlist* 表示兴趣区域集合

1. *SElist*=*null*;//存放轨迹跨越的兴趣区域
2. for each trajectory TR_i in *TD*
   //判断轨迹 TR_i 的起点和终点是否在不同的兴趣区域 Rlist 中
3. if(judge(TR_i,*Rlist*))
4. SElist.add($startRegion_i$,$endRegion_i$);
5. endif
6. endfor
7. for each pair of regions($startRegion_i$,$endRegion_i$) in SElist
   //获取两个兴趣区域间的所有轨迹
8. Tlist=getTrajectory($startRegion_i$,$endRegion_i$);
9. Clusterlist=dbscan(*Tlist*);　//轨迹聚类
   //当簇存在的情况下,才可能有异常轨迹
10. if(Clusterlist ! =null)
11. for each trajectory TR_j in *Tlist*
12. if TR_j is not in *Clusterlist*
13. Output(TR_j);
14. endif
15. endfor
16. endif
17. endfor

}

3　异常轨迹并行检测算法

3.1　并行框架

利用 MapReduce 并行编程模型,在 DATIR 算法的基础上提出了兴趣区域间异常轨迹并行检测算法 PDATIR,PDATIR 算法的基本思想是通过 Hadoop 平台并行地检测不同兴趣区域间的异常轨迹.

PDATIR 算法的具体框架如图 2 所示,首先根据轨迹的起点和终点所在的兴趣区域,将同一对兴趣区域间的所有轨迹划分到同一个计算节点;这样就能让不同的计算节点同时检测不同兴趣区域间的异常轨迹.

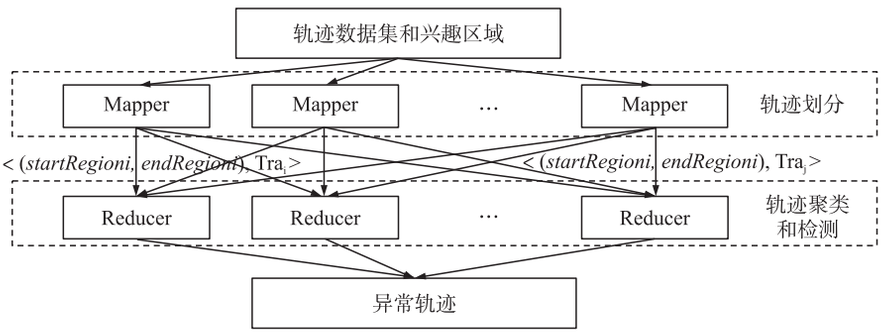

图 2　PDATIR 算法并行框架

Fig. 2　Framework of PDATIR algorithm

3.2　数据分组

PDATIR 算法利用 Hadoop 平台,使得各个计算节点能够同时检测不同兴趣区域间的异常轨迹,并且同一对兴趣区域间的轨迹会分发到同一个节点. 但是,不同兴趣区域间的轨迹数量是不同的,轨迹数量越多,轨迹相似性计算的时间以及聚类的时间会越多. 若只根据兴趣区域 ID 的 Hash 值进行分发,很难保证每个节点检测轨迹所需的时间比较平均.

为了使各个计算节点检测轨迹时间相对比较平均,PDATIR 算法利用贪心思想将兴趣区域进行分组,使得每个计算节点处理的轨迹数量比较均衡,分组算法详见算法 2. 算法 2 的输入数据是轨迹数据集合 *RTlists* 和分组数量 N,其中 *RTlists* 的每个元素存放的是一对兴趣区域中包含的所有轨迹;输出数据则是分组后的轨迹数据集合 *Glists*,其中每个元素存放的是属于同一组的所有轨迹.

算法 2　轨迹分组算法

输入:轨迹数据集合 *RTlists*,分组数量 N

输出:轨迹数据集合 *Glists*

```
divide(RTlists,N,Glists){
  1.Glists=initialize(N);//Glists 元素个数设置为 N
  2.for(i=0;i<RTlists.size;i++)
      //获取存放轨迹数量最少的小组标识 k
  3.    k=Min(Glists);
      //将第 i 对兴趣区域间的轨迹划分到第 k 组
  4.  Add(Glists,RTlists.get(i),k);
  5.endfor
}
```

3.3　并行算法

通过轨迹数据的分组,各个计算节点能够相对均衡地检测兴趣区域间的异常轨迹. 在并行检测阶段,输入的数据是分组后的轨迹数据,每一行数据由分组号 k,起止兴趣区域 ID($startRegion_i$,$endRegion_i$)和对应轨迹集合 *Tlist* 组成. 每个分组可包含若干对兴趣区域及其轨迹,因此,在 map 阶段的主要任务是将每组的轨迹分发到对应的 reduce 计算节点上,reduce 阶段则分别对每组中各兴趣区域间的轨迹进行异常检测. 具体步骤见算法 3.

算法 3　异常轨迹并行检测算法

输入:分组后的轨迹数据集合 *Glists*

输出:异常轨迹

```
map(key,value){
//key 为分组号 k,value 为起止兴趣区域 ID(startRegioni,endRegioni)及其轨迹集合 Tlist
output(k,<(startRegioni,endRegioni),Tlist>);
```

```
}
reduce(key,value){
//key 为分组标识 k,value 为属于该分组标识 k 的所有的兴趣区域间起止兴趣区域 ID(startRegion_i,endRegion_i)及其轨迹集合 Tlist
  1.for each pair(startRegion_i,endRegion_i)
    //获取两个兴趣区域间的所有轨迹
  2.Tlist=getTrajectory(startRegion_i,endRegion_i);
  3.Clusterlist=DBSCAN(Tlist);   //轨迹聚类
  4.   for each trajectory TR_j in Tlist
  5.      if TR_j is not in Clusterlist
  6.         Output(TR_j);
  7.      endif
  8.   endfor
  9.   endfor
}
```

map 过程将均衡分组的标识 k 作为 key 值,轨迹的起点和终点所在的兴趣区域,即轨迹的起止兴趣区域标识符($startRegion_i$,$endRegion_i$)和经过这两个兴趣区域的所有轨迹的集合 $Tlist$ 作为 $value$ 值.

reduce 函数输入的 key 值为分组标识 k,$value$ 值为两个兴趣区域的标识符($startRegion_i$,$endRegion_i$)和经过这两个兴趣区域的所有轨迹的集合. 对两个兴趣区域间的所有轨迹进行 DBSCAN 聚类,不同的簇代表不同的路径,不在簇中的轨迹与大部分轨迹的行驶路径不同,即为异常轨迹.

4　实验结果及分析

实验的云计算环境共有 23 台服务器(1 个主节点,22 个从节点),每个节点配置相同. 节点的处理器为 Intel(R)Xeon(R)CPU E5-2620 v2,主频为 2.10 GHz,操作系统为 64 位 Red-Hat-Enterprise-Linux-Server-release-6.5-(Santiago)操作系统,Hadoop 版本为 hadoop-2.3.0. 本实验采用的数据集为 2012 年 11 月份北京约 1 万 2 千辆出租车 30 天内采集到的数据[15],采样点的时间间隔约为 1 min,总大小为 50 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 200 辆、400 辆、600 辆和 800 辆出租车的轨迹用于实验测试,其数据大小分别为 115 MB、239 MB、429 MB、660 MB.

图 3 为 DATIR 算法的部分检测结果示例,图中标注轨迹的起始兴趣区域 A 为北京天坛公园附近,终止兴趣区域 B 为北京西站附近. 三角形和矩形标注的轨迹路径分别代表两个簇,表示大部分移动对象行驶的轨迹,而圆形标注的轨迹不在任何簇内,标记为异常轨迹. 从图 3 可以明显看出,圆形标注的轨迹存在明显的绕路行为.

图 3　异常轨迹检测结果示例

Fig. 3　An example of experimental result

图 4 为 DATIR 算法和 PDATIR 算法在不同数据集下的运行时间,其中并行算法 PDATIR 算法在 Hadoop 平台上的计算节点数量为 20 个. 从图 4 可以看出,DATIR 算法的运行时间明显多于 PDATIR 算法,并且随着轨迹数据量的增大,两种算法运行时间的差距也越来越大. PDATIR 算法利用 Hadoop 平台并行计算的特点,很大程度上提高了算法的效率.

如图 5 所示,利用 660MB 的轨迹数据集,测试 PDATIR 算法在不同计算节点下的运行时间. 从图中可以看出,随着计算节点数量的不断增加,算法的运行时间逐渐减小,这说明 PDATIR 算法具有较高的加速比. 当节点数量大于 15 时,算法的运行时间减小得不再明显,这是由于随着计算节点的增加,并行平台通信时间占总花费时间的比例也在增加.

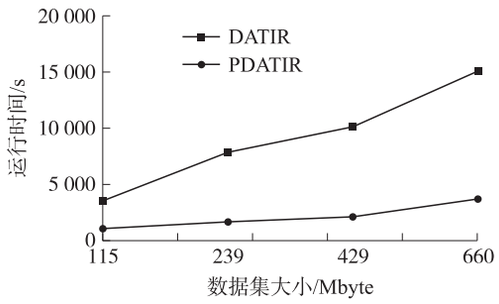

图 4　算法 DATIR 和 PDATIR 运行时间比较

Fig. 4　The running time comparison of two algorithms

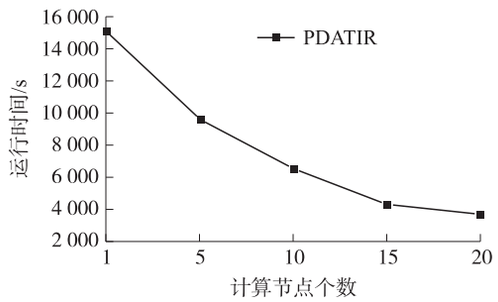

图 5　算法 PDATIR 的加速比

Fig. 5　The speedup ratio of algorithm PDATIR

5　总结

本文从兴趣区域的角度提出一种基于聚类的异常轨迹检测算法,该算法综合考虑了轨迹的局部特征和全局特征,能够有效地挖掘兴趣区域间的正常路径,并能检测兴趣区域间的异常轨迹. 兴趣区域通常是移动对象相对集中活动的区域,检测兴趣区域间异常轨迹能够发现异常行为,并且能够为交通规划提供决策. 为了进一步提高算法的检测效率,利用 MapReduce 框架实现了异常轨迹并行检测算法,用于大规模异常轨迹检测.

[参考文献]

[1] 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.

[2] 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.

[3] 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. California,USA:ACM,2007:330-339.

[4] ZHENG Y. Trajectory data mining:an overview[J]. ACM transactions on intelligent systems and technology(TIST),2015,6(3):29.

[5] GUPTA M,GAO J,AGGARWAL C,et al. Outlier detection for temporal data[J]. Synthesis lectures on data mining and knowledge discovery,2014,5(1):1-129.

[6] 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.

[7] 安计勇,朱猛,翟靖轩,等. 轨迹多因素异常集成检测[J]. 计算机工程与设计,2015,36(10):2700-2705.

[8] 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(ICDE). Cancún,México:IEEE,2008:140-149.

[9] 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.

[10] ZHANG D,LI N,ZHOU Z H,et al. iBAT:detecting anomalous taxi trajectories from GPS traces[C]//Proceedings of the 13th international conference on Ubiquitous computing. Beijing,China:ACM,2011:99-108.

[11] 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,2010:1733-1736.

[12] FONTES V C,de ALENCAR L A,RENSO C,et al. Discovering trajectory outliers between regions of interest[C]//GeoInfo 2013:Proceedings of the XIV Brazilian Symposium on Geoinformastics. Jordão,São Paulo,Brazil:GeoInfo,2013:49-60.

[13] 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(ICPR). Hong Kong:IEEE,2006,3:1135-1138.

[14] WANG Y,LEI P,ZHOU H,et al. Using DTW to measure trajectory distance in grid space[C]//Proceedings of the 4th IEEE International Conference on Information on Science and Technology(ICIST). Shenzhen,China:IEEE,2014:152-155.

[15] 数据堂. 某北方城市 12000 辆出租车 GPS 位置数据(2012 年 11 月)[DB/OL]. [2013-9-16]. http://www.datatang.com/data/44502.

[责任编辑:陆炳新]