# 基于轨迹信息熵分布的异常轨迹检测方法*

蒋 华，郑依龙†，王 鑫
（桂林电子科技大学 计算机与信息安全学院，广西 桂林 541004）

**摘 要**：针对异常轨迹检测多特征检测和检测单元造成的检测效率低等问题，提出一种基于轨迹信息熵分布的异常轨迹检测方法。该方法根据轨迹偏转角与速度将轨迹分割成若干轨迹段，计算轨迹段间加权多特征距离判断轨迹间相似度，进而完成轨迹聚类并计算出每类代表性轨迹，然后对待检测轨迹进行分割，利用代表性轨迹计算每个轨迹段的信息熵，通过比较轨迹信息熵大小及其分布特点实现异常轨迹检测。大西洋飓风数据仿真实验结果表明，该方法提高了聚类效果，克服以整条轨迹检测效率低的缺点，提升了异常轨迹检测算法的有效性。

**关键词**：信息熵；相似度；轨迹聚类；代表性轨迹；异常检测
**中图分类号**：TP391 **文献标志码**：A **文章编号**：1001-3695(2018)06-1655-05
doi:10.3969/j.issn.1001-3695.2018.06.012

## Trajectory outlier detection based on trajectory information entropy distribution

Jiang Hua, Zheng Yilong†, Wang Xin
(*School of Computer & Information Security, Guilin University of Electronic Technology, Guilin Guangxi* 541004, *China*)

**Abstract**: In view of fact that the detection efficiency of the multi-feature detection and detection unit for trajectory outlier is inefficient, this paper proposed a new method named TOD-TIED(trajectory outlier detection based on trajectory information entropy distribution). Firstly, the algorithm partitioned a trajectory into a set of trajectory segments according to corner and velocity, then calculated the weighted multi-feature distance to determine the similarity between trajectory segments. Finally, it grouped trajectories into clusters and calculated representative trajectory. The algorithm partitioned the trajectory into a set of trajectory segments, then calculated the information entropy of each trajectory by using the representative trajectory, finally it detected the abnormal trajectory detection according to the trajectory information entropy and its distribution characteristic. The simulation results of Atlantic hurricane data show that this method can improve the clustering effect and overcome the shortcomings of inefficient detection with the whole trajectory, and improves the effectiveness of the outlier trajectory detection algorithm.

**Key words**: information entropy; similarity; trajectory cluster; representative trajectory; outlier detection

## 0 引言

随着 GPS 定位、传感网络、RFID 等技术日臻成熟以及定位设备的广泛使用催生了大量的轨迹数据，轨迹数据表现为定位设备所产生的大规模数据流。对数据流形式的轨迹数据进行分析和处理，可以发掘出轨迹数据中潜藏的异常现象，从而应用于服务城市规划、气象监测[1]、交通运输[2,3]和热点区域发现[4]等。目前，轨迹数据研究主要包括轨迹聚类[5,6]、轨迹频繁模式以及异常检测等。轨迹异常检测旨在从轨迹数据集合中发掘出偏离正常运动模式的轨迹，有利于为现实问题作出良好决策，如气象监测，提前预测飓风行驶路径或是否发生突然变化，对提前做好预防措施起到重要的作用。

常用的异常轨迹检测算法主要包括轨迹特征抽取、轨迹相似度检测以及分类器等。其中，Lee 等人[7]提出的 TRAOD 算法是经典的异常轨迹检测算法，该算法分为轨迹划分和异常检测两个阶段，该算法需要预计算有关参数，存在适用性低、结果准确性不高等问题；刘良旭等人[8]提出以轨迹点表示局部特征的异常点检测算法 TraLOD 算法，异常检测过程需要执行轨迹子序列间大量计算，导致时间和空间开销较大；文献[9]提出一种基于轨迹大数据离线挖掘与在线实时监测的出租车异常轨迹检测算法，将离线轨迹数据与实时轨迹相结合进行异常轨迹检测，该算法的优点是具有实时性，但是阈值设置困难，若设置不当将会影响检测效果。传统异常检测方法中包括基于聚类的方法，将不属于任何类的对象视为异常对象，同时能够获取到有意义的类簇。而现有聚类算法在轨迹检测方面主要是用于挖掘轨迹频繁模式，而异常轨迹检测方面研究中使用得较少。目前很多轨迹聚类算法被提出，常见的是采用对轨迹分段后再对轨迹进行形状上聚类，相比以整条轨迹聚类方式，更有利于局部特征的比较，如 TRACLUS[10]方法。为了提高异常轨迹检测效率，文献[11]提出了 MapReduce 异常轨迹检测并行算法，该算法采用并行计算的方式提高了计算效率，但在异常轨迹检测过程没考虑轨迹的其他特征(如速度、方向和偏转角)的影响，因此，不能提高异常轨迹检测效果。

针对上述异常轨迹检测存在的问题，提出了基于轨迹信息熵分布的异常轨迹检测方法。该方法分为轨迹段聚类和异常检测两个阶段。在轨迹段聚类阶段，借鉴了文献[12]中定义的二阶段拐点算法思想，在轨迹分割中加入速度限制条件对轨

**收稿日期**：2017-03-20；**修回日期**：2017-04-27 **基金项目**：2016 广西高校中青年教师基础能力提升项目(ky2016YB150)

**作者简介**：蒋华(1963-)，男，河南信阳人，教授，博士，主要研究方向为数据库系统、信息安全；郑依龙(1989-)，男(通信作者)，安徽安庆人，硕士研究生，主要研究方向为信息安全、异常检测(yilong694474390@163.com)；王鑫(1976-)，男，陕西蓝田人，副教授，硕士，主要研究方向为无线传感器网络协议.

迹进行分割,然后将本文提出的加权多特征距离用于度量轨迹段间距离,最后采用改进的密度聚类方法实现轨迹段聚类,进而利用代表性轨迹产生算法[10]求得每个类簇的代表性轨迹,为异常检测中轨迹信息熵计算提供依据。在异常检测阶段中,引入轨迹信息熵概念,并结合统计判别法检测异常数据的"$k\sigma$"准则确定异常轨迹阈值,最后通过比较轨迹信息熵大小及其分布特点实现异常轨迹检测。为了验证本文算法的异常检测效果,本文基于真实数据集与 TRAOD 算法进行实验对比。实验结果表明本文提出的基于轨迹信息熵分布的异常检测算法提高了聚类效果,提升了异常检测的有效性。

## 1 相关定义

### 1.1 轨迹

**定义 1** 轨迹(trajectory)是多维度空间下有序的点的集合。$TR=\{P_1,P_2,P_3,P_4,P_i,\cdots,P_n\}$ $(1\leqslant i\leqslant n)$,其中 $P_i$ 是多维度点,$n$ 是轨迹长度即轨迹所包含的采样点的数目,对于不同轨迹 $n$ 值可能不一样。$TR$ 是由若干个轨迹段组成的。轨迹段形式化表示为 $\text{subTR}=\{P_{j1},P_{j2},P_{j3},P_{j4},P_{ji},\cdots,P_{jn}\}$ $(1\leqslant jn\leqslant n)$,其中,$1\leqslant j_1<j_2<j_3<\cdots<j_n\leqslant n$。

**定义 2** 偏转角是轨迹上相邻两个子轨迹运动方向上的夹角。已知点 $P_1$、$P_2$、$P_3$ 三个点,由这三个点构成的夹角称为开放角,开放角的补角称为偏转角(图 1 中 $\beta$)。如图 1 所示中 $\beta$ 与 $\theta$ 分别是偏转角与开放角,偏转角表示在该点运动方向发生变化程度,偏转角度越大表明运动发生偏转程度越大。$\beta$ 和 $\theta$ 为

$$\beta=\frac{p_1p_2\times p_2p_3}{|p_1p_2|\times|p_2p_3|}\tag{1}$$

$$\theta=\pi-\beta$$

**定义 3** 速度是轨迹上每个采样点处的移动对象运动的速率(此处的速度是等于距离与时间的比值)。轨迹上采样点 $P_i$ 的速度 $v_i$,计算公式如式(2)所示。

$$v_i=\frac{\sqrt{(x_{p_{i+1}}-x_{p_{i-1}})^2+(y_{p_{i+1}}-y_{p_{i-1}})^2}}{t_{i+1}-t_{i-1}}\tag{2}$$

### 1.2 轨迹异常

轨迹特征中包括位置、速度、方向以及转角等特征信息。轨迹分析要从以上四个特征进行考虑,才能更加全面地得出轨迹的运动模式,发掘出异常运动模式的轨迹。在给出异常轨迹定义前,首先介绍加权多特征距离、轨迹段相似度和轨迹信息熵概念。

**定义 4** 加权多特征距离(weighted multi-feature distance, WMFD)借鉴了文献[13]结构相似度方法来衡量轨迹间相似程度的方法,重新定义了轨迹段间距离。轨迹间的 WMFD 包括四个特征上的距离比较,包括速度上的距离、角度上的距离、位置上的距离以及方向上的距离。加权多特征距离是上述四个距离的加权和,如式(3)所示。

$$\text{WMFD}(L_i,L_j)=[\text{speDis},\text{angDis},\text{locDis},\text{dirDis}]\times W^{\mathrm{T}}\tag{3}$$

其中:$W=[w_1,w_2,w_3,w_4]$,$w_1+w_2+w_3+w_4=1$。根据不同的应用场景适当调节向量 $W$ 各分量的大小,本文中为充分考虑各个特征的影响,各分量都设置为 0.25。为了消除不同特征的量纲对计算的影响,每种距离都是归一化处理后的距离。各个距离计算方法在下面给出。

a)速度上的距离 $\text{speDis}(L_i,L_j)$ 是指两个轨迹段平均速度的差值。通过计算两条轨迹段平均速度差值,得出两条轨迹段整体在运动速度特征上的差异程度。速度上的距离计算公式为

$$\text{speDis}(L_i,L_j)=\left|\frac{1}{N_{L_i}}\sum_{k=1}^{N_{L_i}}v_k-\frac{1}{N_{L_j}}\sum_{k=1}^{N_{L_j}}v_k\right|\tag{4}$$

其中:$L_i$ 和 $L_j$ 分别表示两条不同的轨迹段;$v_k$ 表示轨迹段上的轨迹采样点处的速度。

b)角度上的距离 $\text{angDis}(L_i,L_j)$ 是指两条轨迹内部方向波动变化程度情况,反映了轨迹内部波动状况。角度上的距离计算公式如式(5)所示。

$$\text{angDis}(L_i,L_j)=\left|\sum_{k=1}^{N_{L_i}}\beta_k-\sum_{k=1}^{N_{L_j}}\beta_k\right|\tag{5}$$

c)位置上的距离 $\text{locDir}(L_i,L_j)$ 表示两条轨迹在位置上的差异程度情况,在本文中采用 Hausdorff 距离公式来计算两条轨迹位置上的距离。位置上的距离计算公式如式(6)所示。

$$\text{locDis}(L_i,L_j)=\max(d(L_i,L_j),d(L_j,L_i))\tag{6}$$

其中:$d(L_i,L_j)=\max\limits_{x\in L_i}(\min\limits_{y\in L_j}(\text{dist}(x,y)))$ 为两条轨迹 $L_i$、$L_j$ 的直接 Hausdorff 距离,即 $L_i$ 中的一个点 $x$ 到最近的 $L_j$ 中的最大距离,$\text{dist}(x,y)$ 表示两个点之间的欧氏距离。通过位置上的距离可以得出轨迹段在空间位置上的聚集程度。

d)方向上的距离 $\text{dirDis}(L_i,L_j)$ 是表示两条轨迹段在运动方向上的整体偏转差异程度。如图 2 所示,其中 $\alpha$ 是两条轨迹段夹角。两条轨迹段方向相同则其方向(夹角小于 $\pi/2$)距离为两条轨迹段方向向量夹角正弦与较短轨迹长度乘积,如果方向相差很大(夹角大于 $\pi/2$)则此时轨迹方向相差程度很大,用较长轨迹段长度表示方向上的距离。方向上的距离计算如下:

$$\text{dirDis}(L_i,L_j)=\begin{cases}\min(\|L_i\|,\|L_j\|)\times\sin(\alpha) & 0\leqslant\alpha\leqslant\dfrac{\pi}{2}\\ \max(\|L_i\|,\|L_j\|) & \dfrac{\pi}{2}<\alpha\leqslant\pi\end{cases}\tag{7}$$

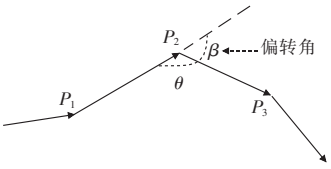



图 1 轨迹偏转角和开放角

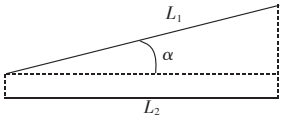



图 2 方向上的距离

**定义 5** 轨迹段相似度(trajectory similarity)是指两条轨迹段间的相似程度。当 WMFD 越小,表示轨迹段越相似,反之越不相似。由定义 4 可以看出,加权多特征距离计算方法体现了轨迹段在多特征上差异程度。

**定义 6** 轨迹信息熵(trajectory entropy)$H(T)$ 是轨迹 $T$ 划分后所有轨迹段的信息熵的累加和。轨迹信息熵计算公式如式(8)所示。

$$H(T)=-\sum_{L_i\in T}^{m}\sum_{L_j\in\text{CTR}}^{n}p(L_i)\log p(L_i)\tag{8}$$

其中:$p(L_i)=\dfrac{\text{dist}(L_i,L_j)}{\sum\limits_{L_k\in\text{CTR}}^{m}\text{dist}(L_i,L_k)}$ 是轨迹 $T$ 中轨迹段 $L_i$ 与代表性轨迹 $L_j$ 的 WMFD 比上该轨迹段与所有代表性轨迹的 WMFD 之和;$\text{dist}(L_i,L_j)=\text{WMFD}(L_i,L_j)$;CTR 是代表性轨迹集合,是利用文献[10]中的代表性轨迹算法计算出聚类后每个类簇得到的。

**定义 7** 异常轨迹(outlier trajectory)是指偏离大多数轨迹运动模式的轨迹。异常轨迹呈现轨迹运动形态(形状)上的异常,或者是运动形态表现为正常,但轨迹其他特征表现异常,如运动速度特征上的异常等。

## 2　异常轨迹检测

为了提高异常轨迹检测的有效性,本文提出基于轨迹信息熵分布的异常轨迹检测方法(trajectory outlier detection based on trajectory information entropy distribution,TOD-TIED),主要包括轨迹段聚类和异常检测。其中,轨迹段聚类部分包括轨迹预处理、轨迹分割以及轨迹段聚类。

### 2.1　轨迹聚类

#### 2.1.1　轨迹预处理

先是轨迹数据集的选取,本文采用了大西洋飓风数据集,从飓风数据提取轨迹运动的时间、经度以及纬度信息,然后根据定义 2 和 3 计算出轨迹的采样点速度和偏转角,为下一步轨迹分割作准备。

#### 2.1.2　轨迹分割

轨迹分割是轨迹段聚类中重要的一步,轨迹分割的合理与否将会直接影响到轨迹段间加权多特征距离比较的准确性,从而进一步影响到轨迹段聚类的有效性和代表性轨迹产生,进而影响轨迹信息熵计算以及异常轨迹检测。文献[7,10,12]在轨迹分割中只是考虑到轨迹的距离和方向上的变化,而忽略轨迹记录点速度变化。由于在轨迹某些位置上轨迹记录点速度变大(变速点),而这些变速点在轨迹上也具有重要价值,如飓风登陆时候速度变化等。通过变速点和偏转角来确定关键点能够比较好地分割轨迹,因此本文在轨迹分割中加入速度限制,实现对轨迹的分割。轨迹分割过程如下:

a)扫描轨迹点序列,$\forall i \in \text{TRlenth}$,如满足 $\Delta v = |v_{i+1} - v_i| \geqslant v_{\max}$,其中 $v_{\max}$ 为给定值,$v_{i+1}$、$v_i$ 分别是轨迹点 $P_{i+1}$、$P_i$ 的速度,则点 $P_i$ 判定为轨迹的待选分割点满足则跳至步骤 b),不满足则扫描下一个轨迹点。

b)判断待选分割点偏转角度,若点 $P_i$ 处的偏转角度 $\beta$,与给定转角阈值 $\delta$,满足 $\beta \geqslant \delta$ 则将判定为备选轨迹还分割点且转至步骤 c),否则转至步骤 a)。

c)根据给定的轨迹划分阈值 $\gamma$ 与比例系数 $k(0 \leqslant k \leqslant 1)$,若满足 $KD_s + (1-k)D_A \geqslant \gamma$ 则将该点作为轨迹分割点并标记该轨迹采样点,其中 $D_s = \Delta v$ 表示速度的变化值,$D_A$ 表示轨迹段在某点处偏转角,通过 $k$ 调节 $D_s$ 和 $D_A$ 在轨迹分割所占的比重大小以适应不同应用环境的需要。重复上述步骤,直到遍历完所有轨迹采样点,算法结束。轨迹分割算法时间复杂度为 $O(n)$,其中 $n$ 是轨迹采样点的个数。

#### 2.1.3　轨迹段聚类

通过研究文献[7,10]发现,基于密度的聚类方法能够发现任意形状的聚类,较好地适应轨迹段聚类,最为重要的是通过调整相关参数来控制聚类的覆盖范围。因此,本文采用密度聚类算法思想来完成轨迹聚类。轨迹段聚类算法中涉及到的相关定义与经典的 DBSCAN 定义类似,对其中部分定义进行了改进。

**定义 8**　轨迹段 $L_i$ 的 $\varepsilon$-邻域 $N_\varepsilon(L_i)$ 定义如下:

$$N_\varepsilon(L_i) = \{L_j \in D \mid \text{WMFD}(L_i, L_j) \leqslant \varepsilon\} \tag{9}$$

其中:$\varepsilon$ 是给定的参数。轨迹段聚类算法中密度可达、直接密度可达、密度相连和核心轨迹段定义与文献[10]TRACLUS 聚类中定义相同。

轨迹段聚类算法描述如下:

输入:轨迹集合 $TR = \{TR_1, TR_2, TR_3, TR_4, \cdots, TR_n\}$;参数 $v_{\max}, \delta, \gamma, k, W, \text{minTra}, \varepsilon$。

输出:轨迹段聚类结果集合 $O = \{C_1, C_2, \cdots, C_m\}$。

```
for each(tr ∈ TR) do
    partition(tr, v_max, δ, γ, k) → newTR; /* 轨迹分割 */
mark all subTR in newTR as unclassified
/* 根据 WMFD 计算轨迹段间距离,求出轨迹段邻域 */
for each L ∈ newTR do
    if(L is unclassified) then
        compute N_ε(L); /* 计算邻域 */
        if(|N_ε(L)| ≥ minTra) then /* 聚类核心轨迹段判定 */
            assign clusterId to ∀X ∈ N_ε(L); /* 设置聚类 ID */
            add N_ε(L) − L into the queue Q;
            /* 扩展轨迹的近邻聚类 */
            expandCluster(Q, clusterId, ε, minTra);
            increase clusterId by 1; /* 初始 clusterID 设为 0 */
        else Mark L noise;
```

上述聚类算法要计算轨迹段的邻域,其时间复杂度为 $O(n)$,该算法整体时间复杂度为 $O(n^2)$。若采用 R-tree 空间索引,时间复杂度降为 $O(n \log n)$。为了更加直观地分析轨迹聚类结果,应用文献[10]中代表性轨迹产生算法得到每个类簇的代表性轨迹,通过代表性轨迹来表示轨迹聚类信息。

### 2.2　异常检测

统计判别法检测异常数据常使用 $k\sigma$ 准则,不需要指定参数,计算方式简便,并且应用广泛,其基本原理如下:

通过定义 6 轨迹信息熵求得待检测轨迹集合的轨迹信息熵集合 $\text{TraEn} = \{e_1, e_2, e_3, \cdots, e_n\}$,将其作为观测数据集,若满足判别式 $|e_i - \bar{e}| > k\sigma$ 的为异常轨迹,其中,$\bar{e} = \frac{1}{n}\sum_{i=1}^{n} e_i$ 是轨迹信息熵平均值,$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(e_i - \bar{e})^2}{n-1}}$ 为标准差。

**引理 1**　若随机变量服从 $\xi \sim N(\mu, \sigma^2)$,则有[14]:

$$P(|\xi - \mu| \geqslant 1.645\sigma) = 0.1$$

通常 $k\sigma$ 准则的参数 $k$ 取值为 2 或 3。根据引理 1,Jiang 等人[14]通过实验证明当数据近似服从高斯分布时,取 $k = 1.645$ 时,$k\sigma$ 准则对异常数据判定较为合理可靠,而本文中的轨迹信息熵数据近似服从高斯分布。因此,本文在异常检测阶段中,取 $k = 1.645$ 作为异常轨迹判定依据。

信息熵对随机事件具有不确定性度量的特点[15]。通过定义 6 的轨迹信息熵将其应用于轨迹异常检测,衡量轨迹所属的运动模式的确定性。若确定性越高,则轨迹信息熵值 $H(T)$ 就越小,则轨迹就越有可能属于现有某种运动模式,即正常轨迹。确定性越低,则轨迹信息熵值 $H(T)$ 越大,表示待检测轨迹偏离大多数轨迹的运动模式,即异常轨迹。而对于那些轨迹信息熵显著高于其他轨迹的轨迹信息熵的轨迹可判定为异常轨迹。故异常检测中的异常轨迹为轨迹信息熵大于 $\bar{e} + k\sigma$。因此,异常轨迹的判断阈值由 $\text{threshold} = \bar{e} + k\sigma$ 计算得到。异常轨迹检测阶段流程如图 3 所示,其步骤总结如下:

a)计算待检测轨迹的采样点处的速度和偏转角。

b)设置参数 $v_{\max}$、$\gamma$、$k$ 和 $\delta$,对轨迹进行分割得轨迹段集合。

c)利用代表性轨迹以及 b)求得轨迹段集合,根据轨迹信息熵定义,得到所有轨迹的信息熵,并计算出轨迹信息熵的均值、标准差以及 threshold。

d)若给定的轨迹信息熵 $e$,满足 $e > \text{threshold}$,则标记该条

轨迹为异常轨迹，否则标记为正常轨迹。

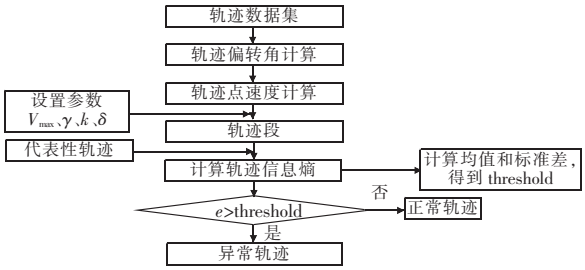



图 3 异常轨迹检测流程

## 3 仿真实验结果与分析

### 3.1 实验数据与环境

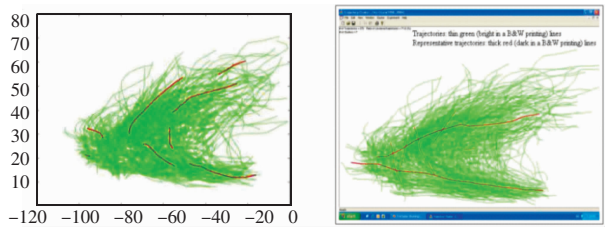
为了验证本文提出的 TOD-TIED 算法性能，进行仿真实验。实验软/硬件环境包括：a）实验环境为 Windows 7，CPU（Intel Core™ i5 2.79 GHz），内存 4 GB，MyEclipse 10，MATLAB 2016；b）实验采用的数据集为大西洋飓风（http://weather.unisys.com/hurricane/atlantic），数据集中包括飓风经度、纬度、中心最大速率、海平面压力、采样时间等信息。本文抽取了 1950 年到 2009 年的数据子集，其中包括 19 750 个采样点组成的 639 条轨迹，并且选取了经度、纬度和时间三个属性信息进行实验。

### 3.2 轨迹聚类实验结果分析

本文算法聚类阶段中涉及到的参数包括偏转角阈值 $\delta$、轨迹分割处阈值 $\gamma$、比例系数 $k$ 和速度阈值 $v_{max}$，调整参数 $\delta$、$\gamma$、$k$ 和 $v_{max}$ 大小将会直接影响分割的轨迹段数目以及计算代价，为了考虑平衡速度和角度对轨迹分割的影响程度，$k$ 值设置为 0.5。从表 1 可以看出，速度参数 $v_{max}$、$\delta$ 越大满足条件的分割点越少，进而轨迹分割后的子轨迹数目也少。而参数 $\gamma$ 越大，则满足条件的轨迹采样点的偏转角度越大，而且在该处的速度变化越大。因此随着 $\gamma$ 增大，满足条件的分割点数目将会减少，进而分割后的轨迹段数目也会减少。参数 $\varepsilon$ 和 minTra 影响轨迹聚类效果，经过多次实验，当 $\varepsilon$ 和 minTra 分别取 0.2 与 30 最为合适。由于本文算法参数的含义清晰明确，针对不同的应用场景，该应用领域的专家比较容易选取合适的参数。因为本文算法的轨迹分割、聚类距离计算与文献[10]的 TRACLUS 算法不同，所以，两个算法不具有直接可比性。故本文仅实验最后部分对最终聚类效果与 TRACLUS 算法进行比较。为了与 TRACLUS 算法进行比较，本文采用了文献[10]TRACLUS 中的代表性轨迹表示方法，红色实线表示聚类的每类代表性轨迹，绿色实线表示其他轨迹，见电子版。图 4 是本文提出的算法（实验参数 $\varepsilon=0.2$，minTra $=30$，$k=0.5$，$W$ 各分量都是 0.25）和 TRACLUS 算法在飓风数据集上的实验结果，通过对比可以发现，本文算法聚类效果更好。由于本文算法在轨迹分割中将轨迹采样点速度特征考虑到其中，选择的轨迹分割点保留了轨迹的局部特征不至于流失。并且由于文献[10]中 TRACLUS 算法距离计算中只考虑到位置关系，而没有从多个特征角度进行轨迹段间相似度计算，导致了将其他特征距离（如速度和轨迹内部角度偏转程度）相差较大的轨迹也被聚成一类。而本文算法是从轨迹特征多个角度考虑，将速度和轨迹偏转角度上距离相差较大的轨迹聚到不同类中。因此，本文采用 WMFD 作为距离度量的聚类算法，相比 TRACLUS 算法具有更好的聚类效果。

表 1 轨迹分割参数

| $V_{max}$ | $\delta$ | $\gamma$ | $N_{subTrajectory}$ | $N_{cluster}$ |
|---|---|---|---|---|
| 20 | 150 | 0.8 | 1 486 | 4 |
| 20 | 150 | 0.85 | 1 345 | 3 |
| 20 | 120 | 0.8 | 1 252 | 8 |
| 30 | 120 | 0.8 | 1 180 | 6 |
| 40 | 120 | 0.8 | 1 154 | 7 |

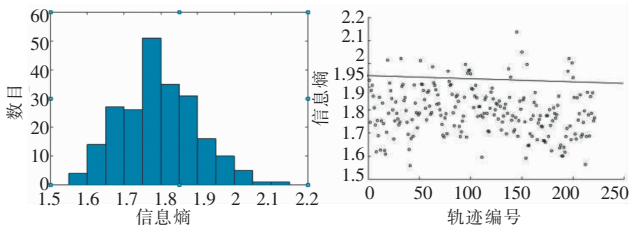


(a)本文实验结果 (b)TRACLUS 实验结果

图 4 飓风数据集下两算法聚类效果比较

### 3.3 与 TRAOD 的异常检测对比与分析

通过计算所有待检测轨迹信息熵并进行统计分析，确定异常轨迹阈值 threshold 为 1.95。从图 5（a）轨迹信息熵频率直方图看出，其总体呈现中间高两边低的形态，绝大数的轨迹信息熵主要分布在（1.65，1.95），其中轨迹信息熵在（1.75，1.8）轨迹数目最多。从图 5（b）散点图分布中也可以看出，少数轨迹信息熵高于 1.95，绝大多数轨迹信息聚集在 1.65 ~ 1.95。实验中轨迹信息熵分布特点符合轨迹运动情况，由于轨迹信息熵是由待测轨迹与每类代表性轨迹计算得到的，而信息熵越高，表明与代表性轨迹运动模式相差越大，则是偏离大多数轨迹的运动模式的异常运动模式。




(a)轨迹信息熵频率直方图 (b)轨迹信息熵散点图

图 5 轨迹信息熵分布

与 TRAOD 对比，由于 TOD-TIED 与 TRAOD 算法距离定义与计算方式，两者计算量差别较大，所以两者在性能上不具有可比性。本文实验对 TOD-TIED 与 TRAOD 在算法效果上以及实用性进行比较与分析。图 6 是 TOD-TIED 与 TRAOD 算法在大西洋飓风数据子集（1990—2006）的检测结果。异常轨迹呈现的运动方向或其他轨迹特征完全不同于相邻轨迹，或者与之相同的轨迹数目很少。图 6（a）和（b）中的粗红色实线表示异常轨迹，见电子版。从图 6 中可以看出，中间区域的异常轨迹的运动方向完全不同于其他轨迹运动方向，以及右侧的异常轨迹完全没有与之相同运动模式的相邻轨迹。图 6（a）显示了本文的 TOD-TIED 异常检测结果，粗红色实线表示的异常轨迹的轨迹信息熵都是高于 threshold 的轨迹，绿色实线表示正常轨迹，见电子版。图 6（b）是 TRAOD 异常检测实验结果（参数设置 $D=85$，$p=0.95$，$F=0.2$）。从图 6 中可以发现，TOD-TIED 与 TRAOD 都能够检测出明显偏离大多数轨迹运动模式的异常轨迹。TRAOD 检测出多数异常轨迹主要是形状上异常以及明显偏离大多数轨迹运动模式的异常。而从图 6（b）可以看出，在 TRAOD 中正常的轨迹在图 6（a）TOD-TIED 实验结果中

是异常轨迹,是因为 TOD-TIED 方法不仅仅除了能检测出形状异常的轨迹,还能够检测出形状正常但是其他特征表现异常的轨迹。如轨迹运动的路径正常,但是轨迹移动的速度异常的轨迹。由于轨迹信息熵距离计算采用加权多特征距离,是从多个角度判断轨迹间相似度,进而能够发掘出轨迹其他特征(如速度等特征)存在异常的隐藏的异常轨迹。

(a)TOD-TIED 实验结果　　(b)TRAOD 实验结果

图 6　TOD-TIED 与 TRAOD 的实验结果对比

另外,在算法参数选取上,本文提出的 TOD-TIED 方法利用异常数据判定的 $k\sigma$ 准则选取异常轨迹阈值,在异常检测阶段无须设置参数,灵活性较高,并且能够全面地了解轨迹运动分布情况,而 TRAOD 需要提前进行相关参数设置,适用性低,且若参数设置不合理将会影响到异常检测效果。因此,本文提出的 TOD-TIED 要比 TRAOD 算法更具有现实意义。

## 4　结束语

本文研究异常轨迹检测,针对异常轨迹多特征检测算法检测效率低的问题,提出了 TOD-TIED。该方法在轨迹段聚类阶段,轨迹段间距离度量应用了加权多特征距离,实验结果表明本文的轨迹段聚类方法发掘出的聚类更具有实际意义。在异常轨迹检测阶段,定义了轨迹信息熵以及采用 $k\sigma$ 准则确定异常轨迹阈值对异常轨迹进行检测。检测结果表明,本文提出 TOD-TIED 算法提高了异常轨迹检测效果,是一种有效的异常轨迹检测算法。由于聚类阶段时间复杂度较高,所以,下一步工作是优化聚类过程,提高聚类的效率。

**参考文献:**


[1] 毛嘉莉,金澈清,章志刚,等. 轨迹大数据异常检测:研究进展及系统框架[J]. 软件学报,2017,28(1):17-34.

[2] Yu Yanwei, Cao Lei, Rundensteiner E A, *et al*. Detecting moving object outliers in massive-scale trajectory streams[C]//Proc of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2014: 422-431.

[3] Liu Siyuan, Ni L M, Krishnan R. Fraud detection from taxis' driving behaviors[J]. IEEE Trans on Vehicular Technology, 2014, 63(1): 464-472.

[4] Chen Zaiben, Shen Hengtao, Zhou Xiaofang. Discovering popular routes from trajectories[C]//Proc of IEEE International Conference on Data Engineering. Washington DC: IEEE Computer Society, 2011: 900-911.

[5] Costa G, Manco G, Masciari E. Dealing with trajectory streams by clustering and mathematical transforms[J]. Journal of Intelligent Information Systems, 2014, 42(1): 155-177.

[6] Deng Ze, Hu Yangyang, Zhu Mao, *et al*. A scalable and fast OPTICS for clustering trajectory big data[J]. Cluster Computing, 2015, 18(2): 549-562.

[7] Lee J G, Han Jiawei, Li Xiaolei. Trajectory outlier detection: a partition-and-detect framework[C]//Proc of IEEE International Conference on Data Engineering. Piscataway, NJ: IEEE Press, 2008: 140-149.

[8] 刘良旭,乐嘉锦,乔少杰,等. 基于轨迹点局部异常度的异常点检测算法[J]. 计算机学报,2011,34(10):1966-1975.

[9] 韩博洋,汪兆洋,金蓓弘. 一种基于轨迹大数据离线挖掘与在线实时监测的出租车异常轨迹检测算法[J]. 中国科学技术大学学报,2016,46(3):247-252.

[10] Lee J G, Han Jiawei, Whang K Y. Trajectory clustering: a partition-and-group framework[C]//Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2007: 593-604.

[11] 唐梦梦,吉根林,赵斌. 利用 MapReduce 的异常轨迹检测并行算法[J]. 地球信息科学学报,2015,17(5):523-530.

[12] Chang Cheng, Zhou Baoyao. Multi-granularity visualization of trajectory clusters using sub-trajectory clustering[C]//Proc of IEEE International Conference on Data Mining Workshops. Washington DC: IEEE Computer Society, 2009: 577-582.

[13] 袁冠,夏士雄,张磊,等. 基于结构相似度的轨迹聚类算法[J]. 通信学报,2011,32(9):103-110.

[14] Jiang Shengyi, Li Qinghua, Li Kenli, *et al*. GLOF: a new approach for mining local outlier[C]//Proc of International Conference on Machine Learning and Cybernetics. Piscataway, NJ: IEEE Press, 2003: 157-162.

[15] Kafsi M, Grossglauser M, Thiran P. Traveling salesman in reverse: conditional Markov entropy for trajectory segmentation[C]//Proc of IEEE International Conference on Data Mining. Piscataway, NJ: IEEE Press, 2015: 201-210.


(上接第 1627 页)


[4] Tu Shitao, Zhu Lanjuan. A bandit method using probabilistic matrix factorization in recommendation[J]. Journal of Shanghai Jiaotong University: Science, 2015, 20(5): 535-539.

[5] Song Yang, Zhuang Ziming, Li Huajing, *et al*. Real-time automatic tag recommendation[C]//Proc of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2008: 515-522.

[6] 王守辉,于洪涛,黄瑞阳,等. 基于模体演化的时序链路预测方法[J]. 自动化学报,2016,42(5):735-745.

[7] Xiong Liang, Chen Xi, Huang T K, *et al*. Temporal collaborative filtering with Bayesian probabilistic tensor factorization[C]//Proc of SIAM International Conference on Data Mining. 2010: 211-222.

[8] Salakhutdinov B R, Mnih A. Probabilistic matrix factorization[C]//Proc of the 20th International Conference on Neural Information Processing Systems. Vancouver, Canada: Curran Associates Inc., 2015: 1257-1264.

[9] Bokde D, Girase S, Mukhopadhyay D. Matrix factorization model in collaborative filtering algorithms: a survey[J]. Procedia Computer Science, 2015, 49(1): 136-146.

[10] Li Gai, Ou Weihua. Pairwise probabilistic matrix factorization for implicit feedback collaborative filtering[J]. Neurocomputing, 2016, 204(9): 17-25.

[11] 田尧,秦永彬,许道云,等. 基于双信任机制的 TrustSVD 算法[J]. 计算机科学与探索,2015,9(11):1391-1397.

[12] 涂丹丹,舒承椿,余海燕. 基于联合概率矩阵分解的上下文广告推荐算法[J]. 软件学报,2013,24(3):454-464.

[13] Amin A, Colman A, Grunske L. An approach to forecasting QoS attributes of Web services based on ARIMA and GARCH models[C]//Proc of IEEE International Conference on Web Services. Washington DC: IEEE Computer Society, 2012: 74-81.

[14] Wang Jian, Deng Wei, Guo Yuntao. New Bayesian combination method for short-term traffic flow forecasting[J]. Transportation Research Part C: Emerging Technologies, 2014, 43(6): 79-94.

[15] Yu Yong, Hui C L, Choi T M. An empirical study of intelligent expert systems on forecasting of fashion color trend[J]. Expert Systems with Applications, 2012, 39(4): 4383-4389.

[16] 孙光福,吴乐,刘淇,等. 基于时序行为的协同过滤推荐算法[J]. 软件学报,2013,24(11):2721-2733.

[17] Koren Y. Collaborative filtering with temporal dynamics[C]//Proc of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2009: 89-97.

[18] 章登义,欧阳黜霏,吴文李. 针对时间序列多步预测的聚类隐马尔可夫模型[J]. 电子学报,2014,42(12):2359-2364.

[19] 王守涛. 一种基于多维时间序列分析的音乐推荐系统研究与实现[D]. 南京:南京大学,2014.