文章编号: 1000 - 4653(2017)03 - 0049 - 05

基于 Spark 的船舶航行轨迹聚类方法

彭祥文[a]， 高 曙[a]， 初秀民[b]， 何 阳[a]， 陆 丛[a]
(武汉理工大学 a. 计算机科学与技术学院;
b. 国家水运安全工程技术研究中心，武汉 430063)

摘 要: 依托船舶自动识别系统(Automatic Identification System，AIS)数据，利用云计算并结合聚类算法，对船舶历史数据进行轨迹聚类分析，构建船舶航行正常轨迹模型，为实时检测船舶异常轨迹奠定基础，进而为提高水上交通监管智能化水平提供新方法。针对目前轨迹聚类算法效率低等问题，基于 Spark 内存计算技术及数据分区思想，提出一种改进的并行子轨迹聚类算法 SPDBSCANST(Parallel DBSCAN of Sub Trajectory Based on Spark)。以长江航道武汉段船舶航行数据为例进行试验验证，并通过可视化方式呈现。结果表明，改进后的算法的聚类效率和效果都有明显提升。

关键词: 水路运输; 船舶自动识别系统; Spark; 轨迹聚类; 正常轨迹建模
中图分类号: U675.7　　**文献标志码:** A

Clustering Method of Ship's Navigation Trajectory Set Based on Spark

PENG Xiangwen[a]， *GAO Shu*[a]， *CHU Xiumin*[b]， *HE Yang*[a]， *LU Cong*[a]
(a. School of Computer Science and Technology; b. National Water Transportation Safety Engineering Technology Research Center, Wuhan University of Technology, Wuhan 430063, China)

Abstract: Constructing normal navigation trajectory model through processing historical AIS(Automatic Identification System) data of ships with the trajectory clustering algorithm is a way of setting up the reference for real-time detection of abnormal ships trajectory. Aimed at the problem of low efficiency of the current trajectory clustering algorithm, an improved parallel sub trajectory clustering algorithm is proposed named as SPDBSCANST (Parallel DBSCAN of Sub Trajectory Based on Spark) featuring Spark memory computing technology and data partition. The algorithm is verified with the ship navigation data of Yangtze River Waterway. The visualization of the trajectories is also achieved. The experiments show that the efficiency of the improved clustering algorithm is increased significantly.

Key words: waterway transportation; AIS; Spark; trajectory clustering; normal trajectory modeling

近年来，随着国内水运业迅速发展，长江干线的交通压力日益增大，迫切需要提高水上交通监管的智能化水平。因此，依托船舶自动识别系统(Automatic Identification System，AIS)数据，基于 Spark 云平台，采用数据挖掘技术，对船舶航行轨迹进行聚类分析，构建正常轨迹模型，为发现和研究船舶运动特征及行为模式提供新思路。

现有的轨迹聚类算法[1]主要分为以下2大类:

1) 将整条轨迹作为研究对象进行聚类。该类方法能比较直观地评价轨迹间的相似性，受输入参数的影响较小，但对复杂的轨迹容易忽略局部异常信息，且对高维轨迹数据的聚类效果欠佳。

2) 对复杂轨迹进行划分，将子轨迹作为聚类目标。该方法能很好地识别轨迹的局部特征，有效处理高维轨迹数据，结合基于密度的 DBSCAN 聚类算法发现任意形状的轨迹簇，但随着数据规模的增大，DBSCAN 算法会因消耗大量的 I/O 而造成聚类效率低下。

对此，结合数据分区思想和 Spark 云平台高效并行的优势，提出一种改进的基于轨迹分区预处理的并行化子轨迹聚类算法 SPDBSCANST(Parallel DBSCAN of Sub Trajectory Based on Spark)。

收稿日期: 2017-04-25
基金项目: 国家自然科学基金(51479155); 城市灾害地图可视化方法研究(JD20150301)
作者简介: 彭祥文(1992—)，男，江西上饶人，硕士生，研究方向为云计算应用。E-mail: 616456468@qq.com
高 曙(1967—)，女，安徽芜湖人，教授，研究方向为数据挖掘及应用、智能交通。E-mail: gshu418@163.com

1 子轨迹划分及相似性度量方法

1.1 基于 AIS 数据的船舶轨迹提取

受 AIS 设备自身及外界条件的限制[2]，通过 AIS 设备获得的轨迹数据需经过一系列预处理才可采用。将解码后的 AIS 数据上传到 HDFS，使用 Spark 的 filter 算子选取一定范围及一段时间内的 AIS 数据，依据船舶水上移动通信业务标识码(Maritime Mobile Service Identity，MMSI)，按时间顺序提取出船舶轨迹。使用该方法提取出的轨迹通常会出现以下情况:

1) 区域内存在多个往返。采用的解决方法是将 MMSI 相同的船舶轨迹分为多个轨迹，主要依据的是轨迹点之间的时间间隔。船舶在航行时，其 AIS 数据更新间隔一般不会超过 10 min; 而对于折返情况，其时间间隔通常远大于 10 min。因此，可将往返轨迹划分为多个轨迹。

2) 轨迹点位置偏移。计算轨迹点与其前后轨迹点之间的时间间隔及距离间隔，若该轨迹点与其前后点之间的时间间隔较小、距离间隔较大，而其前后点之间的时间间隔较小、距离间隔在正常范围内，则可将该轨迹点作为位置偏移点去除。

1.2 子轨迹划分

船舶在内河航行时，受内河形状、宽度和深度等自身条件及桥梁、风等周围环境的影响，其航行轨迹和航速都会发生变化。通过设置船舶转向角阈值及速度变化率阈值，对船舶轨迹进行划分，其中船舶转向角是指相邻子轨迹段的航迹向之差(见图 1)。

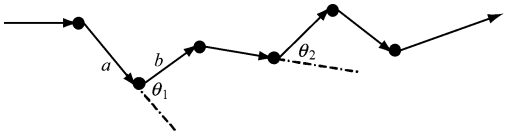


图 1 船舶转向角

图 1 中: a 和 b 为船舶轨迹中相邻的 2 条子轨迹段，其航迹向的夹角(即转向角)为 θ_1。

速度变化率 α 的计算式为

$$\alpha = \frac{|v_2 - v_1|}{\Delta t} \tag{1}$$

式(1)中: v_2 和 v_1 为相邻轨迹点航速; Δt 为相邻时间间隔。

子轨迹划分主要步骤:

1) 计算相邻子轨迹段航迹向差值及相邻轨迹点速度变化率。

2) 将所求值与预先设定的阈值相比较。

3) 若航迹向差值或速度变化率大于阈值，则使用该轨迹点对轨迹进行划分; 否则返回步骤 1)，继续采样。

1.3 子轨迹相似性度量

船舶 AIS 数据中蕴含有丰富的信息[3]，在度量子轨迹的相似性时，应充分考虑各类信息对子轨迹相似性的影响，从而提高聚类质量。这里主要从船舶位置、航向和航速等 3 个方面进行距离计算[1,4]，并通过归一化加权求和得到子轨迹多特征距离，以此度量子轨迹之间的相似性。

1.3.1 子轨迹间位置与航向距离计算

船舶轨迹划分后可表示为子轨迹的集合。在进行轨迹划分时考虑轨迹段航迹向的变化，因此将划分后的子轨迹近似作为线段进行处理。

图 2 为子轨迹间距离度量，其中: $L_i = s_i e_i$ 和 $L_j = s_j e_j$ 分别为 2 条子轨迹; s_i 和 s_j 分别为子轨迹 L_i 及 L_j 的起点; e_i 和 e_j 分别为子轨迹 L_i 及 L_j 的终点; p_s 和 p_e 分别为 s_j 及 e_j 在 L_i(或 L_i 延长线)上的投影。

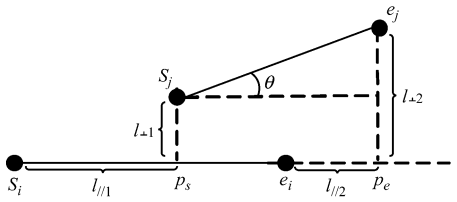


图 2 子轨迹间距离度量

$d_{//}(L_i, L_j)$，$d_{\perp}(L_i, L_j)$ 和 $d_{\theta}(L_i, L_j)$ 分别为子轨迹 L_j 到 L_i 的水平距离、垂直距离及航向距离，具体计算式为

$$d_{//}(L_i, L_j) = \min(l_{//1}, l_{//2}) \tag{2}$$

$$d_{\perp}(L_i, L_j) = \frac{l_{\perp 1}^2 + l_{\perp 2}^2}{l_{\perp 1} + l_{\perp 2}} \tag{3}$$

$$d_{\theta}(L_i, L_j) = \begin{cases} 1 - \cos\theta, & 0° \leqslant \theta \leqslant 90° \\ 1, & 90° < \theta \leqslant 180° \end{cases} \tag{4}$$

同理，可求得子轨迹 L_i 到 L_j 的水平距离 $d_{//}(L_j, L_i)$ 及垂直距离 $d_{\perp}(L_j, L_i)$。根据 Hausdroff 距离定义，取二者中的较大值作为轨迹间的距离。即将子轨迹 L_i 与 L_j 之间的水平距离 $d_{//}$，垂直距离 $d_{\perp}$ 及航向距离 d_{θ} 定义为

$$d_{//} = \max(d_{//}(L_i, L_j), d_{//}(L_j, L_i)) \tag{5}$$

$$d_{\perp} = \max(d_{\perp}(L_i, L_j), d_{\perp}(L_j, L_i)) \tag{6}$$

$$d_{\theta} = d_{\theta}(L_i, L_j) = d_{\theta}(L_j, L_i) \tag{7}$$

1.3.2 子轨迹间航速距离计算

船舶在内河航行时，受内河航道条件的限制，航行轨迹都比较固定，因此船舶航速是轨迹聚类的一

个非常重要的要素。在现有的轨迹聚类算法中,通常只考虑平均航速,对航速信息的利用较少,从最大航速、最小航速、中位数航速及平均航速等 4 个方面综合考虑航速距离的度量。其计算方法为

$$d_S(L_i,L_j) = \frac{1}{4}(S_{\max}(L_i,L_j) + S_{\text{avg}}(L_i,L_j) + S_{\min}(L_i,L_j) + S_{\text{med}}(L_i,L_j)) \quad (8)$$

式(8)中: $S_{\max}(L_i,L_j) = |V_{\max}(L_i) - V_{\max}(L_j)|$ 为 2 个子轨迹中轨迹点最大航速的差异值; S_{avg}, $S_{\min}$ 和 S_{med} 分别为平均航速、最小航速及中位数航速的差异值。

1.3.3 综合距离

在得到 4 种距离的度量方法之后,首先分别对 4 种距离进行归一化处理,然后定义相应的权重 $W=\{W_{//},W_{\perp},W_{\theta},W_S\}$,权重应满足:

(1) 均 >0,即非负性;

(2) $W_{//}+W_{\perp}+W_{\theta}+W_S=1$。

在定义权重时,在不同的内河航道条件及外部环境中所取的权重可以不同,例如: 在较宽的航道,子轨迹间允许的垂直距离会增大,可减小 $W_{\perp}$。由于 4 种距离的量纲不同,因此在计算综合距离之前需对 4 种距离进行归一化,归一化公式为

$$d' = \frac{d - d_{\min}}{d_{\max} - d_{\min}} \quad (9)$$

式(9)中: d 为处理前距离; $d_{\max}$ 和 $d_{\min}$ 分别为该类距离的最大值及最小值; d' 为处理后距离。由此,对 4 种归一化后的距离进行加权求和即可得到综合距离,即

$$D_{\text{dist}}(L_i,L_j) = W_{//}\cdot d_{//}' + W_{\perp}\cdot d_{\perp}' + W_{\theta}\cdot d_{\theta}' + W_S\cdot d_S' \quad (10)$$

2 SPDBSCANST 聚类算法

在采用 DBSCAN 算法对数据进行聚类时,大量的 I/O 消耗导致时间剧增。[5] Spark 分布式云平台引入弹性分布式数据库 RDD(Resilient Distributed Dataset)的概念[6],在计算中将数据分布式缓存在各节点内存中,从而降低大量的磁盘 I/O 消耗。基于 Spark 实现并行子轨迹 DBSCAN 聚类算法,首先对轨迹数据进行分区预处理,分别对各分区子轨迹进行聚类; 然后对各邻近区域进行类簇合并,从而得到最终的轨迹聚类结果。由于 Spark 所有的计算都在内存中对 RDD 进行计算,中间无需与磁盘进行 I/O,因此能极大地提高聚类效率。SPDBSCANST 聚类算法总体流程见图 3。

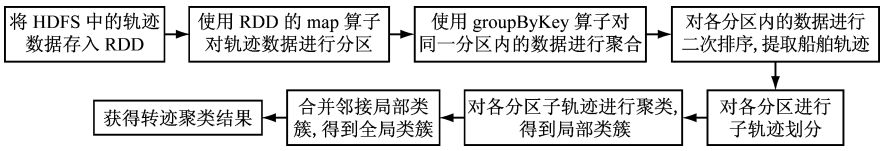


图 3 SPDBSCANST 聚类算法总体流程

SPDBSCANST 聚类算法伪代码描述如下:

SPDBSCANST 聚类算法

算法名称: SPDBSCANST 聚类算法

输入: (1) 邻域 ε,密度阈值 min S_{tr};

(2) 轨迹数据,各分区经度范围;

(3) 分区距离权重 $W\{W_{//},W_{\perp},W_{\theta},W_S\}$

输出: 全局轨迹类簇

BEGIN

1. rdd = sc.textFile(hdfs 文件路径) //将轨迹数据存入到 rdd
2. rdd.map(d = >(num,d)) //依据分区经度的范围对轨迹数据进行划分,num 为划分后分区号
3. rdd.groupByKey() //按分区号聚合轨迹数据
4. rdd.map(BinOrderKey(_)) //对各分区内数据进行二次排序,提取船舶轨迹
5. rdd.map(seprate(_)) //分区子轨迹划分
6. rdd.map(DBSCANST(_)) //子轨迹 DBSCAN 聚类
7. rdd.map(c = >(cnum,c)).reduceByKey() //合并邻接子轨迹类簇

END

2.1 轨迹数据分区处理

轨迹数据的分区可看作是对轨迹的初次子轨迹划分。在进行轨迹数据划分时,由于内河环境复杂,不一定依据经度值(长江在纬度上可看成一条曲线)均匀划分,可根据内河特征进行划分,将轨迹划分为桥梁区域、支流区域和弯道区域等。轨迹分区完成后,采用"1.2"节中的子轨迹划分方法对各区域内的轨迹进行划分。

2.2 分区子轨迹聚类

采用 DBSCAN 聚类算法对各分区子轨迹进行

聚类[8]，使用式(10)度量子轨迹的相似性，依据分区特征，利用分区权值代替全局权值，从而提高聚类质量。子轨迹 DBSCAN 聚类方法与典型的 DBSCAN 聚类方法类似，不同之处在于距离的度量方法。子轨迹 DBSCAN 聚类方法使用的距离为子轨迹对象之间的距离，而典型的 DBSCAN 聚类方法使用的距离为点对象之间的距离。邻域为 ε，密度阈值为 $\min S_{tr}$ 的子轨迹 DBSCAN 聚类算法相关定义如下。

1) 核心对象：给定子轨迹 L_i 的 ε 邻域内的子轨迹数目大于或等于密度阈值 $\min S_{tr}$，具体定义为

$$N_\varepsilon(L_i) = \{L_j \mid D_{\text{dist}}(L_i, L_j) \leqslant \varepsilon\} \quad (11)$$

$$N_\varepsilon(L_i) \geqslant \min S_{tr} \quad (12)$$

2) 直接密度可达：对于子轨迹集合 $\boldsymbol{D}_{TD}$，若子轨迹 L_i 在 L_j 的邻域 ε 内，且子轨迹 L_j 为核心对象，则称子轨迹 L_i 为 L_j 直接密度可达。

3) 密度可达：对于子轨迹集合 $\boldsymbol{D}_{TD}$，若存在子轨迹链 $L_1, L_2, \cdots, L_n$，对于 $L_i \in \boldsymbol{D}_{TD}$ $(1 \leqslant i \leqslant n)$ 存在 L_{i+1} 从 L_i 关于 ε 和 $\min S_{tr}$ 直接密度可达，则称 L_n 为 L_1 密度可达。

4) 密度相连：若存在子轨迹 L_k，使得子轨迹 L_i 和 L_j 都从 L_k 密度可达，则称 L_i 和 L_j 密度相连。

2.3 局部类簇合并

在进行区域划分时，可将原本在全局中为同一类簇的子轨迹类簇划分成 2 个局部类簇(见图 4)[9]。

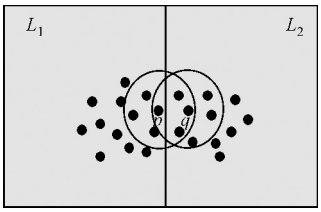


图 4 轨迹类簇合并

图 4 中，黑点代表子轨迹，p 和 q 两条子轨迹同时属于分区 L_1 及分区 L_2 中的类簇，因此可对类簇进行合并。具体合并方法为：

1) 确定划分边界邻接区域，若子轨迹中存在轨迹点在邻接区域内，则将该子轨迹划分到邻接区域内。

2) 遍历邻接区域内所有的子轨迹，若存在子轨迹为核心对象且同时属于 2 个局部类簇，则合并该局部类簇。

2.4 船舶航行轨迹建模

经过以上聚类过程即可得到船舶子轨迹类簇，在各子轨迹类簇中提取一系列采样点(用 SP 表示采样点)表征船舶典型轨迹。以下为船舶航行轨迹建模过程。

2.4.1 确定各子轨迹类簇的方向(簇向)

取各子轨迹类簇中所有轨迹点航向的平均值作为簇向，具体计算方法为

$$COU_w = \frac{COU_1 + COU_2 + \cdots + COU_n}{n} \quad (13)$$

2.4.2 沿着对应簇向划分网格

沿着对应簇向对子轨迹类簇进行网格划分(见图 5)。

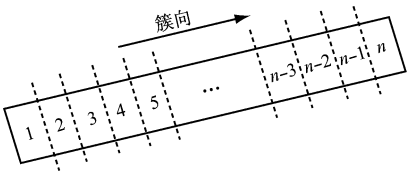


图 5 类簇网格划分

图 5 中：矩形框表示子轨迹类簇；箭头方向表示簇向；n 为类簇划分后的块数，即该子轨迹类簇采样点个数。n 的值通过对类簇内所有完整轨迹(同一 MMSI)的轨迹点总数取平均确定，计算式为

$$n = \frac{\sum_{i=0}^{m} n_{\text{um}P_i}}{m} \quad (14)$$

式(14)中：$n_{\text{um}P_i}$ 为第 i 条完整轨迹中轨迹点个数；m 为完整轨迹数。

2.4.3 构建采样点

图 5 中，每个网格构建 1 个采样点 SP_i，采样点有 4 个特征属性，分别为平均经度 LON_{avg}，平均纬度 LAT_{avg}，平均航速 SPD_{avg} 和平均航向 COU_{avg}，具体表示为

$$SP_i = \{LON_{\text{avg}}, LAT_{\text{avg}}, SPD_{\text{avg}}, COU_{\text{avg}}\} \quad (15)$$

使用采样点表征船舶典型轨迹，具体表示为

$$TR = \{SP_1, SP_2, \cdots, SP_n\} \quad (16)$$

3 试验及分析

试验在武汉理工大学国家水运安全工程技术研究中心的 Spark 云服务平台上完成，创建 6 台虚拟机组成一个集群。处理器配置：8 核；内存 8G；硬盘 300G。软件环境选择 CentOS 系统；Spark1.6.1；Hadoop2.6.4；IDEA3.4；Scala2.10.8；可视化工具使用 Mapv。选取一台虚拟机作为主节点 master，其余为工作节点 worker。试验分为改进后算法对聚类效率的提升和聚类效果的展示 2 部分。

3.1 Spark 云平台下轨迹聚类效率分析

选取长江航道武汉段 2016 年 2 月份的 AIS 数据作为试验数据。为在不同数据量下对算法的效率进行对比，分别选取大约 500M(1 000 万条预处理

后 AIS 数据,只包含纬度、经度、速度、方向、MMSI 及时间) 和 2G 的数据量进行试验,结果见图 6。

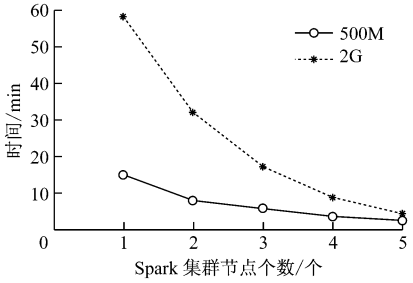


图 6 算法执行时间对比

从图 6 中可看出: 随着集群节点个数的增加,算法执行时间缩短,最后趋于平稳; 数据量越大,算法的加速比越高,从而说明改进后的算法对大数据具有很好的适应性。

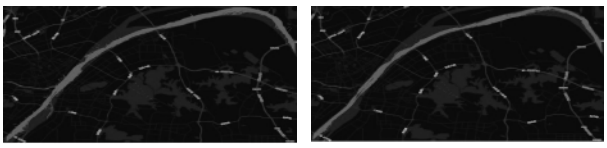
由此可见,利用 Spark 云平台能有效提高海量 AIS 数据的处理效率,数据量越大,效果越明显,从而为高效、大规模地进行船舶航行轨迹分析奠定基础。

3.2 聚类效果展示

选取长江航道武汉段 2016 年 2 月份大船(船长 > 80 m) 的 AIS 数据作为试验数据,经度值在 [114.23°,114.56°],纬度值在 [30.447°,30.73°],对数据进行预处理之后,有效 AIS 数据为 1 948 581 条; 将轨迹数据划分为 20 个区域,对各分区进行子轨迹划分,划分后子轨迹有 96 566 条。该部分试验主要分为 3 部分进行,分别为邻域 ε 及密度阈值 $\min S_{tr}$的确定、不同航道条件下综合距离权值的确定和典型轨迹提取。

3.2.1 邻域 ε 及密度阈值 $\min S_{tr}$的确定

在距离权值(如式(10) 所示) 相等(都为 0.25) 的情况下,对轨迹间距离进行统计,结果表明轨迹间距离大多集中在(0 ~ 0.01) 范围内,故取邻域 ε = 0.01。由于船舶轨迹受航道限制,故轨迹间相似性都比较高,经过多次试验后,当密度阈值 $\min S_{tr}$ = 20 时,聚类结果比较理想。图 7 为船舶轨迹聚类前后对比。



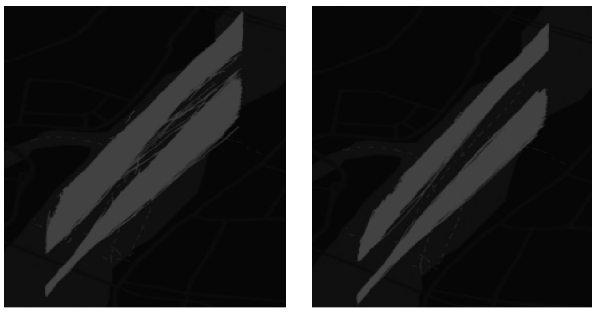
a) 聚类前船舶轨迹 b) 聚类后船舶轨迹

图 7 船舶轨迹聚类前后对比

3.2.2 综合距离权值的确定

从图 7a) 中可看出,对所有分区使用相同的距离权值时,一些分区内的聚类结果不尽如人意(图 8a) 和 9a) 为放大后的 2 个分区) ,因此需基于航道特征确定各距离权值。综合距离从垂直距离、平行距离、角度距离及速度距离等 4 方面考虑,依据航道特征将航道划分为限速区域(桥区,港口等) 、限宽区域(宽航道/窄航道) 及弯道区域。

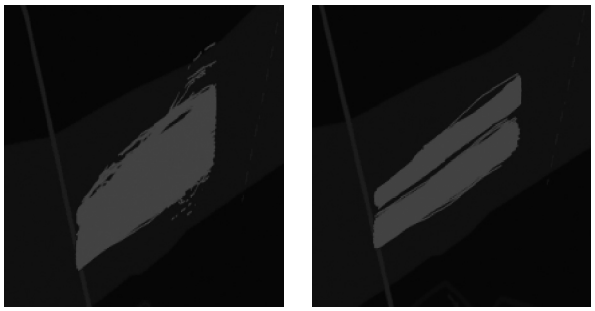
由图 8a) 可知,该航道内有武汉长江大桥、长江二桥及汉江汇流,因此该区域内船舶的航速会受到限制,增大航速距离权值将加大航速对轨迹聚类的影响; 图 8b) 为修改权重 $\boldsymbol{W}$ = (0.2,0.2,0.2,0.4) 后的聚类效果,可发现修改权值后聚类效果有明显提升。



a) 修改前 b) 修改后

图 8 限速区域权值修改前后聚类效果对照

由图 9a) 可知,该航道为夹水道,航道较窄,在该区域内船舶间垂直距离受到限制,故增大垂直距离权值,从而增大垂直距离对聚类效果的影响; 图 9b) 为修改权值 $\boldsymbol{W}$ = (0.15,0.4,0.2,0.25) 后聚类效果,可发现聚类效果有较大改善。



a) 修改前 b) 修改后

图 9 限宽区域权值修改前后聚类效果对照

3.2.3 典型轨迹提取

在确定各分区距离权值之后,采用“2.4”节给出的方法构建船舶航行轨迹。图 10 为船舶典型轨迹提取,其中黑点为提取出的 2 条典型轨迹(分别为上行和下行) 。

(下转第 68 页)

的风险评价法避免了渔业安全监督管理机构仅从事故起数、经济损失等方面描述安全状况的片面性;在考虑多项指标评价各类事故的后果的同时,避免了传统多指标评价方法复杂的操作步骤;此外,还可从事故类型的角度分析渔业安全状况。但是,该方法也存在一定的不足。例如,在第一次进行渔业安全状况评价时,必须根据多年的事故数据严格求取各类事故的相对危害度,这样得到的评价结果才更合理、更科学。

参 考 文 献

[1] 姚杰,任玉清,吴兆麟.渔业安全综合评价模型研究[J].中国航海,2010,33(4):71-74.

[2] 任玉清,郑吉辉,董晖.基于集对分析的渔船安全综合评价模型[J].中国航海,2012,35(2):60-63.

[3] 范耀天.内河船舶交通安全评价法——危险指数法[J].交通科技,2000(3):53-54.

[4] MENTES A, AKYILDIZ H, YETKIN M, et al. A FSA Based Fuzzy DEMATEL Approach for Risk Assessment of Cargo Ships at Coasts and Open Seas of Turkey [J]. Safety Science, 2015, 79(12): 1-10.

[5] 任玉清,姚杰,张飞成,等.安全检查表法在渔船救生设备安全评价中的应用[J].渔业现代化,2015,42(1):72-75.

[6] 任玉清,姚杰,赵希波,等.基于 FTA 的渔船海损事故分析[J].中国航海,2007(1):68-71.

[7] 农业部渔业局.渔船水上事故统计规定:农渔发[2010]41 号[S].2010.

[8] 孙世荃.危险度的概念与应用[J].中华放射医学与防护杂志,1993,13(4):283-285.

[9] 张圣坤.船舶与海洋工程风险评估[M].北京:国防工业出版社,2003.

[10] 张斌.不确定性信息处理的集对论思想与方法[J].模糊系统与数学,2001,15(2):89-93.

[11] 汪新凡.基于模糊语言评估和联系数的多属性群决策方法[J].数学的实践与认识,2007,37(15):54-59.

[12] 李德顺,许开立,叶海云.论基于多元联系数的集对分析评价模型[J].中国安全生产科学技术,2009,5(4):110-114.

[13] 邱菀华.管理决策熵学及其应用[M].北京:中国电力出版社,2011:12-123.

(上接第 53 页)

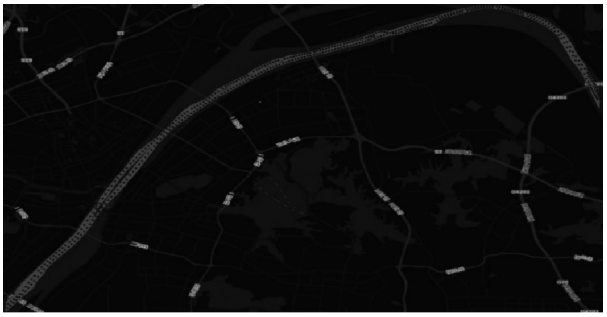


图 10 船舶典型轨迹提取

4 结束语

基于 Spark 云平台,对船舶子轨迹聚类方法进行研究,构建船舶航行轨迹,并以长江航道武汉段 2016 年 2 月份的 AIS 数据为试例进行验证。通过在 Spark 云平台上对船舶子轨迹聚类算法进行并行化设计,可极大地提高轨迹聚类效率,为进一步研究船舶运动特征、行为模式及船舶轨迹实时异常检测等提供技术保障。

参 考 文 献

[1] 肖潇,邵哲平,潘家财,等.基于 AIS 信息的船舶轨迹聚类模型及应用[J].中国航海,2015,38(2):82-86.

[2] 魏照坤.基于 AIS 的船舶轨迹聚类与应用[D].大连:大连海事大学,2015.

[3] 刘畅.船舶自动识别系统(AIS)关键技术研究[D].大连:大连海事大学,2013.

[4] LIU B, DE SOUZA E N, MATWIN S, et al. Knowledge-Based Clustering of Ship Trajectories Using Density-Based Approach [C] // IEEE International Conference on Big Data. IEEE, 2014: 603-60.

[5] 赖丽萍,聂瑞华,汪疆平,等.基于 MapReduce 的改进 DBSCAN 算法[J].计算机科学,2015(S2):396-399.

[6] 王桂兰,周国亮,萨初日拉,等.Spark 环境下的并行模糊 C 均值聚类算法[J].计算机应用,2016,36(2):342-347.

[7] 朱飞祥,张英俊,高宗江.基于数据挖掘的船舶行为研究[J].中国航海,2012,35(2):50-54.

[8] DAI B R, LIN I C. Efficient Map/Reduce-Based DBSCAN Algorithm with Optimized Data Partition [C] // IEEE Fifth International Conference on Cloud Computing. IEEE Computer Society, 2012: 59-66.

[9] SARAZIN T, AZZAG H, LEBBAH M. SOM Clustering Using Spark-MapReduce [C] // Parallel & Distributed Processing Symposium Workshops. IEEE, 2015.