AIS 航线分析

田展源 2020302131226

2022.4.17

摘要

随着水路交通的日益繁华和船舶自动识别系统（AIS）的广泛普及，社会亟待一种能够处理并应用 AIS 航线数据的方法。本文的工作重心就是如何利用好 AIS 数据，并从中获取有价值的信息。

首先，由于 AIS 数据本身存在飘逸和缺漏等干扰性错误，本文第一要务应是优化 AIS 数据质量，使之能够更适应后续的分析。针对噪点问题，本文将噪点分为停靠漂移点、距离漂移点、角度漂移点三种，并根据其特征分别设置噪点判断标准，据此对早点进行剔除。而针对缺漏问题，考虑到船只的速度和船首向都是连续变化的，其轨迹应当连续且光滑，因此，对于满足缺漏条件的数据段，本文采用三次样条插值对此部分轨迹进行拟合，并对质量优化结果进行评估。

其次，由于 AIS 航线轨迹数据量较大，且存在冗余。本文对质量优化后的数据进行表头归并压缩和曲线压缩两大压缩处理。其中，由于轨迹中数据点的船只信息、代码信息、时间信息（年份和月份）存在巨大重复记录，因此本文首先将大部分数据点都拥有的信息提取并归并到表头，作为前提信息记录，即表头归并压缩，这一部分也是无损压缩。而考虑到数据点记录量远大于描绘大致曲线所需的点数，因此本文对曲线进行压缩，采用道格拉斯-普克法，去除对曲线弯曲程度贡献不大的点，进而完成了对曲线的压缩存储，这一部分是有损压缩。压缩完成后，本文分析了不同曲线压缩阈值下的数据损失量和压缩效率，以此找到了最佳曲线压缩阈值（$10^{-3} \sim 10^{-2.5}\mathrm{m}$）。

在对轨迹信息处理完毕后，本文运用轨迹数据信息得到了不同时段下船舶的航行轨迹，并用矢量叉积法和窗检索判断船舶轨迹与长江大桥的交点，并据此得到了 10 月 17 日到 10 月 25 日内不同时间段下长江的船舶交通流量。在认真分析了船舶流量分布后，不难发现流量分布有着很大的时间段相关性。因此，本文选用了时间序列模型对 10 月 26 日各时间段长江船舶流量进行预测，尔后本文先对前九天数据进行平稳化，然后得到了前九天数据的自相关图和偏自相关图，根据两图特征，本文选用时间序列模型中的 ARMA（1，11），并求解其参数。最后顺利得到了 26 日流量数据和其中误差以及置信区间。

至此，本文的工作全部结束。在进行了曲线修复、曲线压缩、数据预测三大工作后，本文最后对采用模型进行评估和分析，以求未来能够更好地工作。

关键字：AIS，漂移点判准，表头归并压缩，道格拉斯-普克法，ARMA

1 问题背景与重述

随着水路交通的日益繁荣，对水上安全监管与调度已经迫在眉睫，而船舶自动识别系统（AIS）应运而生。其获取的大量与船舶有关的静态与动态信息能够自动通过岸基或星载接收站传递给海事系统及船务公司，为有效进行水上交通监控和管理提供了必要的数据支撑。

然而问题也接踵而至，AIS 轨迹数据质量并不理想，有着许多噪点和数据缺失，并且其庞大的数据量给数据传输和数据处理带来很多困难，而在此基础上，AIS 数据的应用领域也是一片空白。因此，本文主要做了以下三大工作：

1. AIS 轨迹数据重建。主要解决了轨迹质量的问题。
2. AIS 轨迹数据压缩。
3. AIS 轨迹数据应用。具体而言，本文根据前若干天的船舶流量情况数据完成了对后续几天船舶流量情况的预测。

2 假设

1. 由于涉及区域较小，本文未考虑地球曲率对距离和角度计算的影响。
2. 由于船舶较多，本文假定各只船均在长江范围内做轨迹平滑的运动。

3 符号说明

符号	意义
V_s	船速
P_{Lon}	P 点经度
P_{Lat}	P 点纬度
T_i	第 i 个点所处时刻
C_r	航线压缩率
$loss$	航线压缩损失

4 AIS 轨迹的质量改善

本文将从噪点去除和数据修复两个方面提高航线的质量。

第一是噪点的去除，而对于 AIS 时空船，可以认为其时间、速度、船首向均为精确数值，而仅考虑坐标噪声，基于此，本文遍历所有数据点，依次判断各个数据点是否为以下三种噪点，并逐个剔除 [5]：

1. 停靠漂移点：$V_s = 0$ 的静止船舶，坐标发生了随机偏移。
2. 距离漂移点：从第 k 点和第 k+1 点船舶前进的速度大于船舶的最大速度，或者船舶在此间段时间内的速度积分，与实际位移距离有明显偏差。
3. 角度漂移点：在行船过程中，AIS 船舶数据点轨迹发生不规则不合理的明显转向。

第二是缺失数据的修复，在行船过程中发生的大面积无数据航线，本文会根据无数据航线段的前后数据点，对航线的可能走向进行预测，进而填补丢失的数据。

4.1 数据预处理

AIS 航线原文档设计 14 个数据项，首先需要做的是对庞大繁杂的数据进行提取，本文根据需要提取出了各个点的时刻点 T_i（T_i =）（s）、坐标 $\{P_{Lon}, P_{Lat}\}$（°）和该处船舶的速度矢量 $\overrightarrow{V_s}$(m/s)（$1kn = 0.5144444m/s$）。然后根据网络墨卡托投影在 WGS84 坐标系下的转化公式 [3]：

$$\begin{cases} x = P_{Lon} * 20037508.34/180 \\ y = \ln\left(\tan\left(90 + P_{Lat}\right) * \pi/360\right) * 20037508.34/\pi \end{cases}$$

可以将经纬度坐标转化为平面直角坐标系坐标 $\{x, y\}$（m），进而求得了点与点之间位移矢量 $\overrightarrow{S}$(m)。

4.2 AIS 轨迹异常点的判断

4.2.1 停靠漂移点

在遍历过程中，首先利用 $V_s = 0$ 条件找到船的停靠点序列 $\{P_i, P_{i+1}...P_{i+n}\}$，调用此停靠点序列各个点的位移矢量并判断 $\left|\overrightarrow{S}\left(x, y\right)\right| > 0.001$（其中 0.001 为设定漂移阈值），若某个中间节点 P_{i+m} 与周围两点的位移矢量均超出漂移阈值，则 P_{i+m} 为漂移点，对其进行删除操作，同时修改前后两端点的链表值。在剔除完所有停靠漂移点后，对剩余的正常停靠点序列进行归并操作：删除所有停靠点，并用其位置均值代替此停靠点的位置。

$$\begin{cases} \overline{x} = \dfrac{\sum x}{n - m} \\ \overline{y} = \dfrac{\sum y}{n - m} \end{cases}$$

4.2.2 距离漂移点

距离飘移点采取以下两种判定方式，P_i 到 P_{i+1} 的位移矢径只要满足任意一种即可认为 P_{i+1} 为漂移点。第一是第 k 点和第 k+1 点船舶前进的速度大于船舶的最大速度：$\left|\overrightarrow{S}\left(x, y\right)\right| > V_{max} * (T_{i+1} - T_i)$，其中的最快船速由于最高限制船速取决于航行所处位置和时间，因此不妨取 $V_{max} = max\left\{V_s\right\}$。第二是 T_i 到 T_{i+1} 时间段内的速度积分与实际矢径长度有巨大差异：$\left|\int_t^{t+1} v_s dt - \left|\overrightarrow{S}\left(x, y\right)\right|\right| > 10$，其中 10m 为设定的漂移阈值。此项判准主要剔除的是未超速，但时间段内速度变化不符合逻辑（比如加速度过大）的距离漂移点，考虑到船舶正常情况多为匀加速运动，因此可化简判断公式。总结果如下。

$$\begin{cases} max\left\{V_s\right\} * (T_{i+1} - T_i) < \left|\overrightarrow{S}\left(x, y\right)\right| \\ \left|\dfrac{Vs_i + Vs_{i+1}}{2} \cdot (T_{i+1} - T_i) - \left|\overrightarrow{S}\left(x, y\right)\right|\right| > 10 \end{cases}$$

4.2.3 角度漂移点

船舶转向能力是船舶机动性的重要参数，一般用摆动直径衡量。船舶设计的最大摆动直径 l 与船舶长度 l 相关，一般取 2 ~ 4 倍船长，即 $d = k * l, k \in [2, 4]$[5]。若船舶速度为 v，以最大摆动直径 d 转向，则转向速率为 $r = \dfrac{360V}{\pi kl}$，此时船舶从轨迹点 i 到轨迹点 i+1 的最大转向角度 $\omega_{\max}$ 应为转向速率在时间上的积分，即

$$\omega_{\max} = \int_{T_i}^{T_i+1} \frac{360v}{\pi kl} dt \leq \frac{360V}{\pi kl} * S_{max}$$

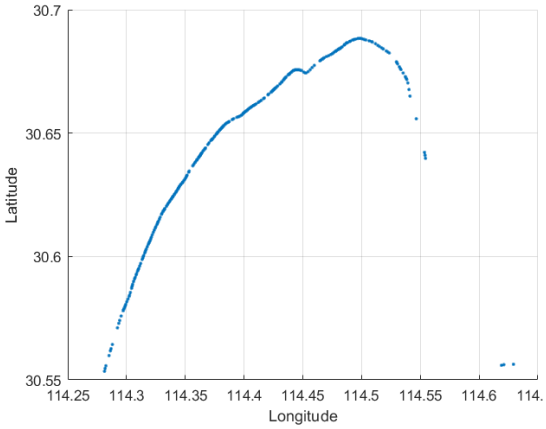

图 1: 原始航线图

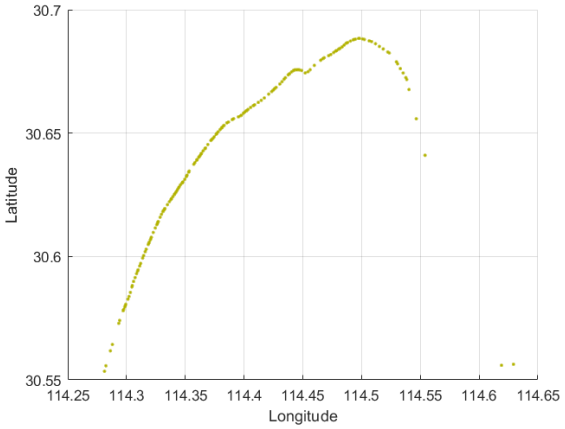

图 2: 去噪后的航线图

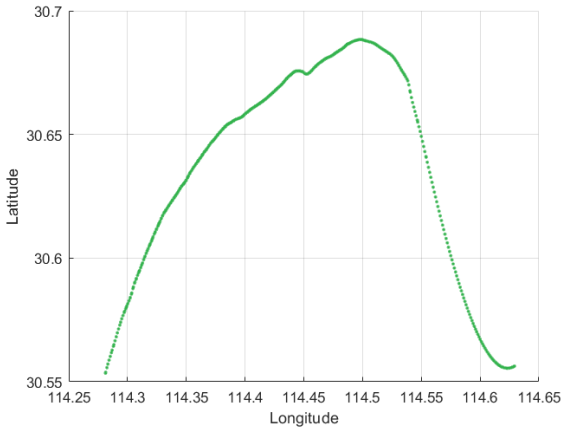

图 3: 修复后的航线图

由 P_i 到 P_{i+1} 的位移矢径可以算出 P_iP_{i+1} 的方向角:$C_i = \arctan \dfrac{S_i(x)}{S_i(y)}$，据此角度漂移点 P_{i+1} 的判断条件为:

$$\arctan \frac{S_{i+1}(x)}{S_{i+1}(y)} - \arctan \frac{S_i(x)}{S_i(y)} > \frac{360V}{\pi kl} * S_{max}$$

4.3 AIS 轨迹缺失点补全

按照河流航线规定，数据点间正常时间间隔应为 3min[5]，故对于已经剔除完噪点的轨迹曲线，对每个 $T_{i+1} - T_i > 180$ 的点 P_i 到 P_{i+1}（缺值区间）进行插值。考虑到船舶航行时的速度和船首向变化都应当是连续的，其轨迹应当处处连续且光滑，并且航行中任何时刻坐标都与前后时刻的坐标高度相关，而与远处坐标关联性相对较弱，本文对每个缺值区间，以此区间为中心，左右各取十个点进行分段三次样条插值（spline）拟合缺失曲线（对于首尾的缺失点，取最前（后）的二十个点进行拟合），即对每个小区间都用一个三次多项式进行拟合，使得整条曲线处处连续且光滑。

4.4 优化成果评估

采用上述方法对各 AIS 航线轨迹图进行质量优化，从视觉观感上效果均十分理想。见**图 1图 2图 3**

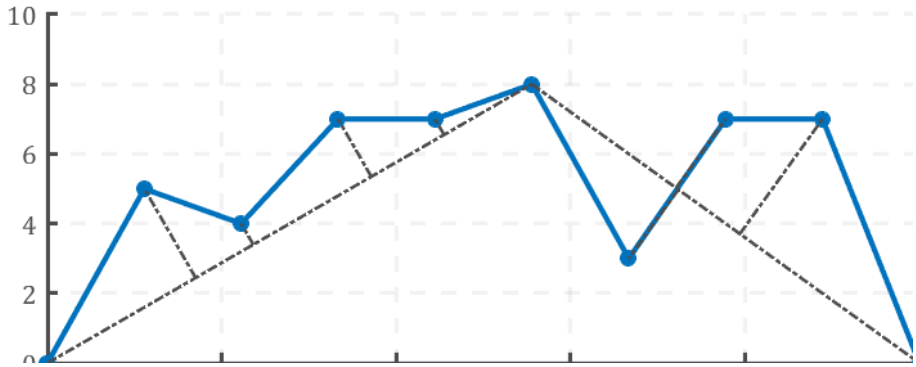

图 4: 道格拉斯普克算法举例

5 AIS 轨迹的压缩存储

对于航线的压缩存储应当分为对曲线轨迹的压缩和对船舶信息的压缩，就曲线压缩而言，即意味着用尽可能少的点对其行走路线进行拟合，因此本文选用曲线压缩算法中的道格拉斯-普克法，并对其加以修改优化，用于压缩存储曲线数据。而对于其他数据，本文采用归并法，将冗余的时间、编码数据用表头的形式进行存储。

5.1 曲线压缩

运用道格拉斯普克算法的基本思想，首先在曲线首尾两点间虚连一条直线，求出其余各点到该直线的距离 l_d。然后选其最大者与阈值 D 相比较，若大于阈值 D，则离该直线距离最大的点保留，否则将直线两端点间各点全部舍去。依据所保留的点，将已知曲线分成两部分处理，重复第 1、2 步进行迭代操作，即仍选 $\max\{l_d\}$ 与阈值 D 比较并依次取舍，直到无点可舍去，最后得到满足给定精度限差的曲线点坐标。

比如在**图 4**所示例子中，第一次连接曲线首尾点后，发现曲线顶点到首位连线距离大于阈值，因此保留顶点，在后续迭代过程中，其余点离首点顶点连线、顶点尾点连线均小于阈值，因此全部可以舍去，最后仅保留下了首点顶点尾点这三个点。

在航线压缩中本文设置阈值 $D = 0.001$，并据此进行航线压缩操作，结果如**图 5**所示，深蓝色曲线为压缩前航线图，浅蓝色曲线为压缩后航线图。

5.2 非曲线信息数据压缩

传统 AIS 数据主要有下列三大冗余：

1. 日期冗余：大部分航线数据点均处于同一天甚至同一小时内，因此可将日期信息单独提取出作为数据表头进行存储，即使偶尔有跨天的船舶，通过将表头日期改成两天也可轻易解决。
2. 前点坐标冗余：在不考虑误差情况下，前一时刻坐标点可由上一坐标点推出，因此也属于冗余信息。
3. 编码冗余：船舶身份码和数据末尾编码在部分情况下也相同，因此与日期同理，可一同归纳入表头进行存储。

而通过表头归并法去除冗余后的文件，数据量直接减少了一半，说明此方法有着良好的压缩效力。值得注意的是，尽管速度和船首向数据在船静止不动的情况下均保持不变，存在大量冗余，但在 AIS

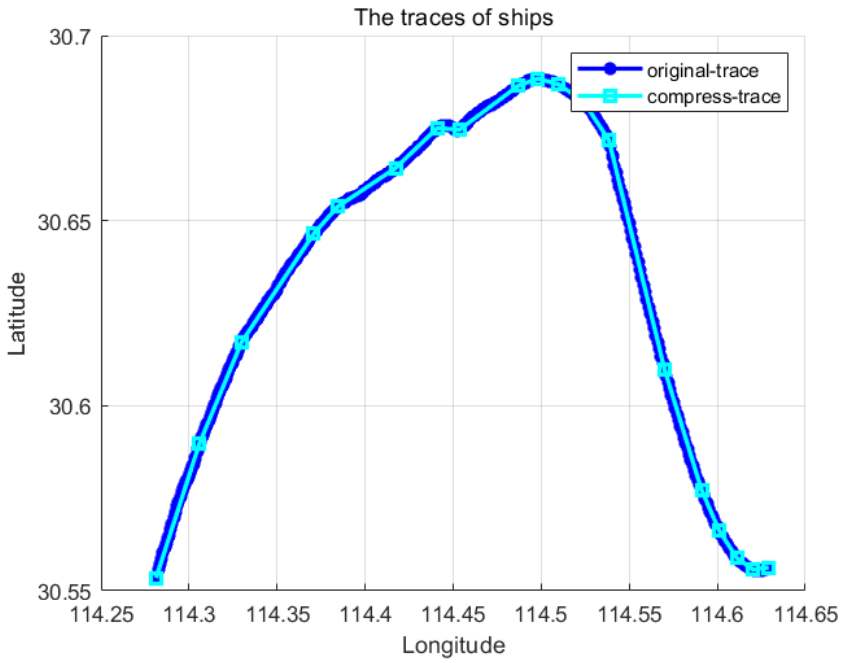

图 5: 压缩前后航线图比较

航线压缩的过程中，本文对停靠漂移点的处理包含了对多于停靠点的归一化和去噪化，因此此冗余无需重复去除。

5.3 压缩质量评估

考虑到采用归并法对船舶信息的表头归并法压缩为无损压缩，本环节质量评定中的压缩损失仅从曲线压缩角度进行评估。具体来说，本文选取了 $10^{-5} \sim 10^{0}$ 共 11 个不同的阈值 D，分别对各种阈值 D 下的压缩比 C_r 和压缩损失率 $loss$ 进行分析，两者的计算公式如下：

$$\begin{cases} C_r = \dfrac{S'}{S_0} \\ loss = \sum |l| \end{cases}$$

式中，S' 代表压缩后的文件大小，S_0 代表原文件大小，两者相除得到的压缩比越低，代表算法压缩效力越高。l_d 代表每次用道格拉斯普克算法曲线拟合过程中舍弃掉的点离拟合折线的距离之和，可用于衡量拟合后折线与原曲线的差别大小。

经过计算，11 个不同阈值 D 的两个指标如**图 6**所示，其中紫色线段代表压缩比随阈值变化的趋势，而橙色曲线代表压缩过程中损失随阈值变化的趋势。可以看到低压缩比总是伴随着较大损失，通过**图 6**也可以找到一个较好的阈值 D（$10^{-3} \sim 10^{-2.5}$），在保证了压缩比的同时，让损失最小化，而不同情况下的最佳 D 则应视具体情况而定，本文不作深入讨论。

6 基于 AIS 轨迹的交通流量预测

6.1 数据预处理

为了得到各个时段中船舶通过长江大桥的船次，本文对去除噪点后的轨迹数据进行遍历，在已知长江大桥两端点经纬度坐标为 $\{114.28335°, 30.55231°\}, \{114.29305°, 30.54697°\}$ 的情况下 [1]，对

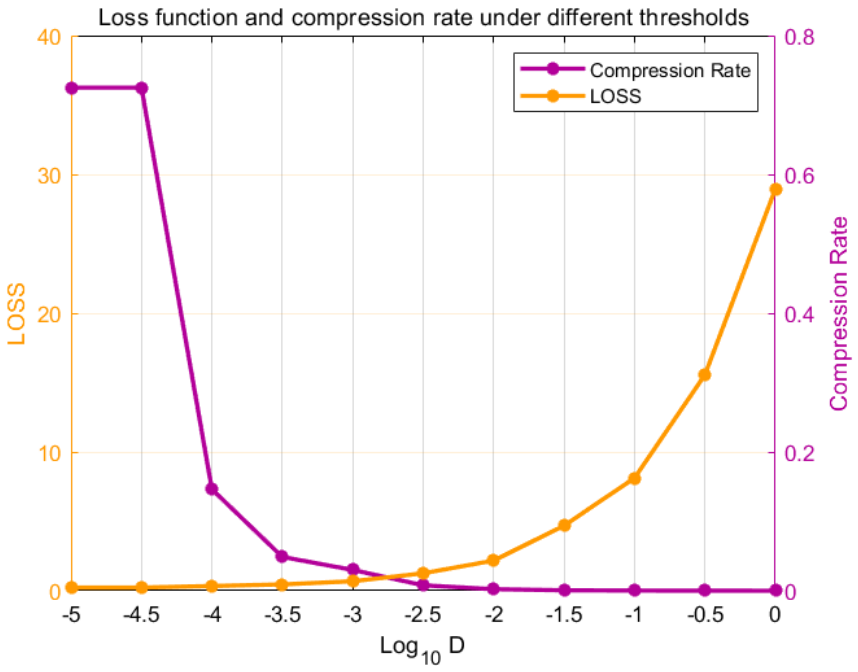

图 6: 压缩前后航线图比较

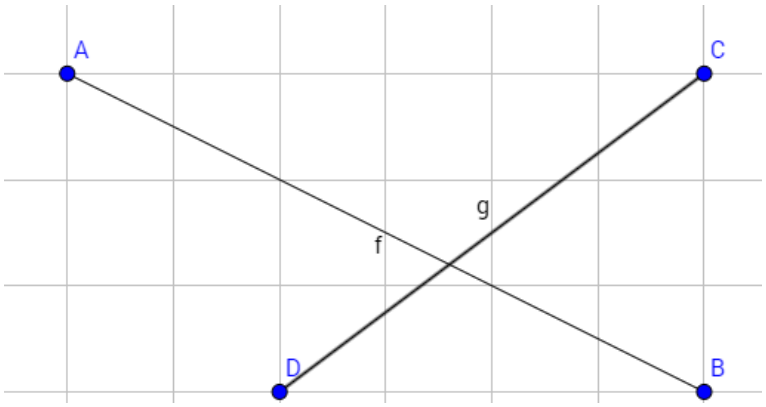

图 7: 矢量叉乘法原理

每个轨迹线段均使用矢量叉积法判断是否相交 [2]:

$$\begin{cases} max(x_C, x_D) > min(x_A, x_B) \\ max(y_A, y_B) > min(y_C, y_D) \\ max(x_A, x_B) > min(x_C, x_D) \\ max(y_C, y_D) > min(y_A, y_B) \\ \overrightarrow{DA} \times \overrightarrow{DC} \cdot \overrightarrow{DB} \times \overrightarrow{DC} < 0 \\ \overrightarrow{AC} \times \overrightarrow{AB} \cdot \overrightarrow{AD} \times \overrightarrow{AB} < 0 \end{cases}$$

其中 x_A 代表点 A 的 x 坐标（经度），$\overrightarrow{AB}$ 代表二维向量 AB。

上述六个不等式同时满足时，线段 AB 与线段 CD 在二维空间内相交。其原理为：判断相交相当于判断 A、B 在线段 CD 的异侧，同时 C、D 也在线段 AB 的异侧，如**图 7**。前者与–$\overrightarrow{DA}$ 转向 $\overrightarrow{DC}$ 的方向和 $\overrightarrow{DB}$ 转向 $\overrightarrow{DC}$ 的方向相反–等价，后者同理可以转化为向量旋转方向的条件，而两个

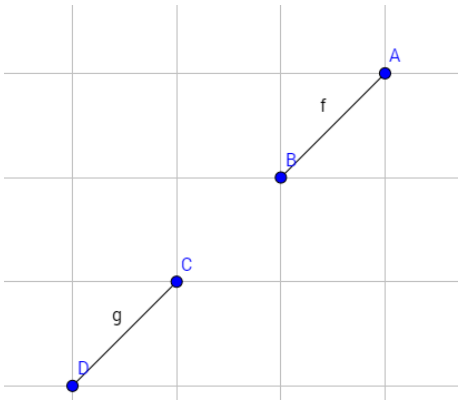

图 8: 矢量叉乘法特殊情况

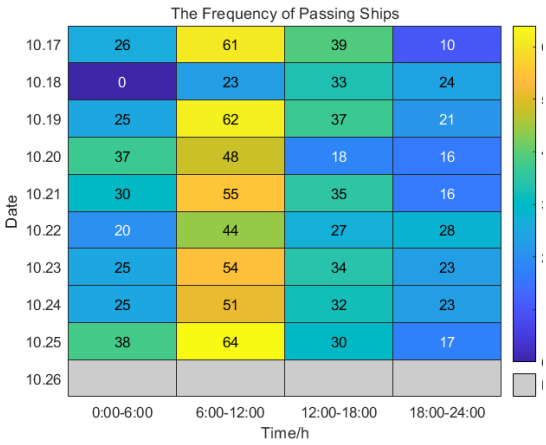

图 9: 前 9 天船舶流量分布

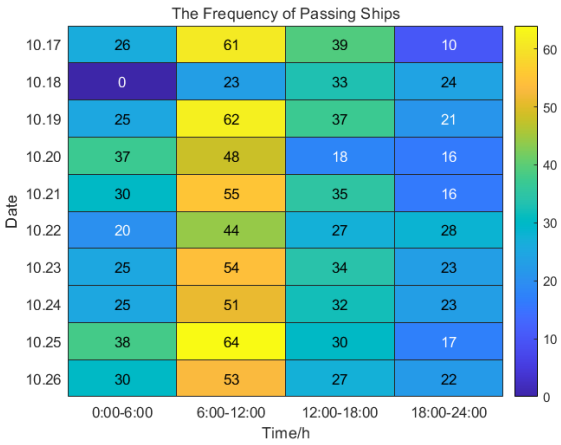

图 10: 10 月 26 日预测数据

二维向量叉积的正与负可以反映两个向量在空间中的相对位置（顺时针或者逆时针），因此可以得到最后两个向量叉乘式。而前四个式子的作用有两个，第一是做初判，可以快速筛去大部分不相交直线，即：窗检索。二是可以排除符合向量叉乘式却不相交直线的情况，如**图 8**所示。

基于此可以得到各个时段通过长江大桥的船舶数量，如**图 9**所示。其中 10.26 日的船舶数量还未进行预测，因此暂时未填入数据。

6.2 时间序列模型的建立与参数求解

通过**图 9**，不难发现船舶流量与时间段关联度很大，比如早上 6 点到 12 点间船舶数量基本是一天中最多的，因此，本文采用时间序列模型预测后续船舶交通情况（时间序列背后的原理较为复杂，详细阐述会占据较多篇幅，因此本文仅描述操作，原理部分可参考 [4]）。

首先研究 10.17 日 0 点到 10.25 日 24 点的船舶通行数量变化曲线，对曲线进行增广迪基–富勒检验，发现曲线并不符合平稳性的单根特征，因此本文先对曲线进行差分以将曲线平稳化。再依次进行增广迪基–富勒检验和白噪声检验，差分后曲线通过了上述两个检验，证明其具有平稳性特征，且不为白噪声，说明曲线具有拟合价值。紧接着，本文对差分后曲线绘制自相关图 **11**和偏自相关图 **12**，可以看到自相关图有着明显的拖尾特征，而偏自相关图在 11 阶后有截尾特征，故本文选用自回归移动平均模型——即 ARMA 模型拟合时间序列：

$$x_t = u_t + \phi_1 u_{t-1} + \phi_2 u_{t-2} + ... + \phi_q u_{t-q} + \theta_1 x_{t-1} + \theta_2 x_{t-2} + ... + \theta_p x_{t-p}$$

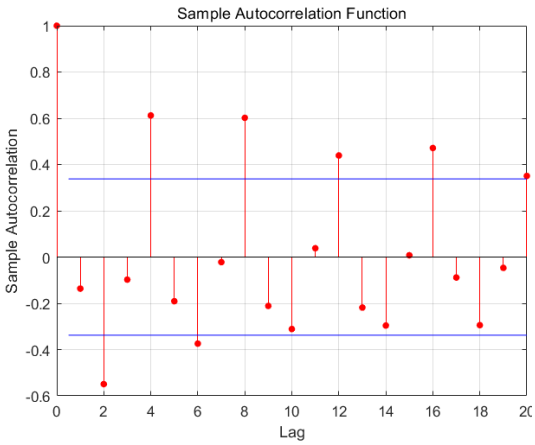

图 11: 差分曲线自相关图

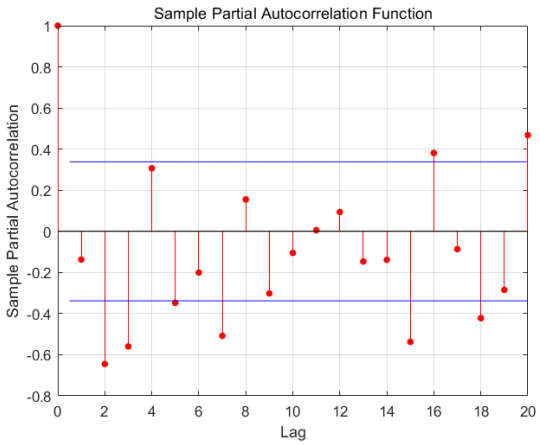

图 12: 差分曲线偏自相关图

ARIMA(11,0,1) Model (Gaussian Distribution):

	Value	StandardError	TStatistic	PValue
Constant	0	0	NaN	NaN
AR{1}	-0.0073273	1.5531	-0.004718	0.99624
AR{2}	-0.82745	0.64538	-1.2821	0.19981
AR{3}	-0.30337	1.4975	-0.20259	0.83946
AR{4}	-0.35123	0.99398	-0.35336	0.72382
AR{5}	-0.64735	0.97221	-0.66586	0.5055
AR{6}	-0.23701	1.3563	-0.17475	0.86128
AR{7}	-0.72679	0.90079	-0.80684	0.41976
AR{8}	0.29015	1.4705	0.19732	0.84358
AR{9}	-0.42455	0.2816	-1.5076	0.13165
AR{10}	0.079294	0.72404	0.10952	0.91279
AR{11}	0.063132	0.27525	0.22936	0.81859
MA{1}	-0.3404	1.5827	-0.21507	0.82971
Variance	49.367	14.369	3.4357	0.000591

图 13: ARMA 参数结果

其中 x_t 代表待预测的 t 时刻船舶通行量，其由 q 个误差项 $\phi_i u_{t-i}$ 和 p 个前端数据影响项 $\theta_j x_{t-j}$，而根据之前对自相关图 **11**和偏自相关图 **12**的分析我们可以预测 $q = 11$,$p = 1$，在此条件下求解各个参数，结果如 **13**所示。最后结果见表 2和图 **10**。

表 2: 10 月 26 日预测数据

0:00-6:00	30
6:00-12:00	53
12:00-18:00	27
18:00-24:00	22

7 模型评价与改进

1. 本文的数据恢复模型能够去除潜在噪点，并完整地还原航行轨迹，但当航线中出现巨大面积数据缺失时，仅靠前后轨迹坐标进行插值是缺少科学性的。

2. 本文的压缩模型能在保持数据质量的情况下最大化压缩比例。

3. 本文预测模型是基于时间序列的预测模型，因此在未来数据的预测上具有较强的科学性和普适性，对其的进一步改进可以加入深度学习框架进行优化，可以让成果质量进一步提高。

参考文献

[1] NAN. Map. https://ditu.google.cn/. 2022.

[2] Siannodel. Csdn. https://blog.csdn.net/qq826309057/article/details/70942061. 2017.

[3] slandarer. Csdn. https://blog.csdn.net/shaxiaozilove/article/details/54908569?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2\%7Edefault\%7ECTRLIST\%7ERate-1.pc_relevant_antiscanv2&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2\%7Edefault\%7ECTRLIST\%7ERate-1.pc_relevant_antiscanv2&utm_relevant_index=2\%EF\%BC\%89. 2020.

[4] 孙建秋. 基于深度学习 CNN-LSTM 的船舶轨迹预测研究 [D]. 宁波大学, 2020.

[5] 张黎翔, 朱怡安, 陆伟, 文捷, 崔俊云. 基于 AIS 数据的船舶轨迹修复方法研究 [J]. 西北工业大学学报, 2021,39(01):119-125.

附件

```
    %以下代码用于得到船舶流量数组
Path = 'C:\Users\tzy\Desktop\大二下\数赛\选拔题\数据\';          % 设置数据存放的文件夹路径
File = dir(fullfile(Path,'*.txt')); % 显示文件夹下所有符合后缀名为.txt文件的完整信息
FileNames = {File.name}';       % 提取符合后缀名为.txt的所有文件的文件名，转换为n行1列
Length_Names = size(FileNames,1); % 获取所提取数据文件的个数
Passing=zeros(10,24)
    %穿过长江大桥的日期和时间段（比如（1，20）就代表17号19-20点间通过大桥的船只数）
bridge=[114.28335,30.55231;114.29305,30.54697]';
for k = 1 : Length_Names
    % 连接路径和文件名得到完整的文件路径
    K_Trace = strcat(Path, FileNames(k));
    [f,message]=fopen(K_Trace{1,1}, 'r');

    %以下代码可用于创建k个数据包，挺好用的
    % 读取数据（因为这里是.txt格式数据，所以直接用load()函数）
    %eval(['Data',num2str(k),'=','load(K_Trace{1,1})',';']);
    % 注意1: eval()函数是括号内的内容按照命令行执行，
    %       即eval(['a','=''2','+','3',';'])实质为a = 2 + 3;
    % 注意2: 由于K_Trace是元胞数组格式，需要加{1,1}才能得到字符串
    [DATA,count] = fscanf(f,'%d-%d-%d %d:%d:%d %d %f %f %f %f %f %f %*s',[13,inf]);
    for i=1:count/13-1
        if(DATA(12,i)==0)%速度为零就跳过
            continue;
        end
        if(iscro(bridge(:,1),bridge(:,2),DATA(8:9,i),DATA(8:9,i+1)))
            Passing(DATA(3,i)-16,DATA(4,i)+1)=Passing(DATA(3,i)-16,DATA(4,i)+1)+1;

        end
    end
    fclose(f);
end


load Passing!.mat
%热图显示

Passing=[sum(Passing(:,1:6),2),sum(Passing(:,7:12),2),sum(Passing(:,13:18),2),sum(Passing(:,19:24),2)];
Passing(10,:)=[30,53,27,22];
h = heatmap(Passing);
h.CellLabelFormat = '%d';
h.MissingDataColor = [0.8 0.8 0.8];
h.MissingDataLabel = 'No Data';
colormap(gca, 'parula');
xlabel("Time/h");
ylabel("Date");
title("The Frequency of Passing Ships")
h.YDisplayLabels
```

```
    =["10.17","10.18","10.19","10.20","10.21","10.22","10.23","10.24","10.25","10.26"];
%h.XDisplayLabels
    =["0:00-1:00","1:00-2:00","2:00-3:00","3:00-4:00","4:00-5:00","5:00-6:00","6:00-7:00","7:00-8:00","8:00-9
h.XDisplayLabels =["0:00-6:00","6:00-12:00","12:00-18:00","18:00-24:00"]
%两种x轴分别为24h制和四段制

%采用时间序列模型预测(四段制)
Passing=[sum(Passing(:,1:6),2),sum(Passing(:,7:12),2),sum(Passing(:,13:18),2),sum(Passing(:,19:24),2)];
Passing=reshape(Passing(1:9,:)',[1,36]);
DPassing=diff(Passing);
plot(1:35,DPassing);
figure(1);
autocorr(DPassing);
figure(2);
parcorr(DPassing);
H1= adftest(DPassing)%平稳性检测通过
yanchi=[6,12,18]; %做6.12.18步延迟
H2=lbqtest(DPassing,'lags',yanchi)%白噪声检验通过


ToEstMd = arima('ARLags',1:11,'MALags',1,'Constant',0);%指定模型的结构
[EstMd,EstParamCov,LogL,info] = estimate(ToEstMd,DPassing');%模型拟合
P_Forecast = forecast(EstMd,4,'Y0',DPassing');
%差分还原
for j=1:4
    Passing(36+j)=Passing(36+j-1)+round(P_Forecast(j));
end
```

```
[f,message]=fopen('C:\Users\tzy\Desktop\大二下\数赛\选拔题\数据\20161017 (23).txt', 'r');
if f==-1
    disp (message); %显示错误信息
end

[DATA,count] = fscanf(f,'%d-%d-%d %d:%d:%d %d %f %f %f %f %f %f %*s',[13,inf]);
%m=sin(90/180*pi());%测试下弧度和角度
%m=atan(-1)/pi()*180;
Trace_vector=zeros(2,count/13); %Trace_vector(i):第i个点向第i+1个点矢径

hold on
grid on
ax=gca;
ax.Color=[1,1,1];
ax.XColor=[1,1,1].*.3;
ax.YColor=[1,1,1].*.3;
ax.LineWidth=1.5;
ax.FontName='cambria';
ax.GridLineStyle='--';
xlabel("Longitude")
ylabel("Latitude")
```

```
%用于美化
plot(DATA(8,:),DATA(9,:),'.')

err=0;
time=zeros(1,count/13);%以第一个点为起算点，秒为单位的时间序列

%数据预处理
for i=1:count/13
    %网络墨卡托运用WGS84坐标系转化,船速度
    %
          (https://blog.csdn.net/shaxiaozilove/article/details/54908569?spm=1001.2101.3001.6650.1&utm_medium=di

    if(i>=2)
        time(1,i)=time(1,i-1)+DATA(6,i)-DATA(6,i-1)+60*(DATA(5,i)-DATA(5,i-1))+3600*(DATA(4,i)-DATA(4,i-1));

    end
    if(i==count/13)
        break;%最后一个点矢径设置为（0，0）
    end
    Trace_vector(1,i)= (DATA(8,i+1)-DATA(10,i+1))*20037508.34/180;%经度转x
    Trace_vector(2,i)=log(tan((90+DATA(9,i+1))*pi()/360))/(pi()/180)*20037508.34/180-log(tan((90+DATA(11,i+1))


end
nowSize=count/13-1;

%去除噪点
for i=2:count/13-1%掐头去尾各个点
    if(i>count/13-err-1)%err去除完了后的max点数达到就退出
        nowSize=count/13-err-1;
        break;
    end

    %停靠判准，速度为零的点聚合至一点上，并将粗差点去除
    if(DATA(12,i)==0)
        k=i;
        adsumx=0;adsumy=0;
        while( k<=count/13-err && DATA(12,k)<=0.1)%遍历后续所有速度为0的点
             p.s.因为后面k要-1，所以可以取到count/13-err
            if(abs(Trace_vector(1,k))+abs(Trace_vector(2,k))<=0.0003)%k点矢径小于一个阈值，说明k+1点不是粗差
                adsumx=adsumx+Trace_vector(1,k);
                adsumy=adsumy+Trace_vector(2,k);
                %粗差点直接跳过，反正都会合并
            end
            k=k+1;
        end
        k=k-1;%将k调整到最后一个速度为0的点上
        DATA(8,i)=DATA(8,i)+adsumx/(k-i);
        DATA(9,i)=DATA(9,i)+adsumy/(k-i);
        Trace_vector(:,i)=Trace_vector(:,k);
```

```
        DATA(:,i+1:k)=[];
        Trace_vector(:,i+1:k)=[];
        time(:,i+1:k)=[];
        err=err+k-i;
        if(k>count/13-err-1)%err去除完了后的max点数达到就退出
            nowSize=count/13-err-1;
            break;
        end
    end

    %速度判准，假设仅作匀加速运动
    Theo_dis_back=(DATA(12,i)+DATA(12,i-1))/2*0.5144444*(time(i)-time(i-1));%一节转化为m/s,算出此速度下可走的距
    Theo_dis_fore=(DATA(12,i)+DATA(12,i+1))/2*0.5144444*(time(i+1)-time(i));%i到i+1距离
    if(abs(Theo_dis_back-sqrt(Trace_vector(1,i-1)^2+Trace_vector(2,i-1)^2))>10)%i-1到i实际距离与理论距离极度不符
        DATA(:,i)=[];
        Trace_vector(:,i)=[];
        time(:,i)=[];
        err=err+1;
        continue;
    end
    %转向判准
    if(abs(atan(Trace_vector(1,i)/Trace_vector(2,i))/pi()*180-atan(Trace_vector(1,i+1)/Trace_vector(2,i+1))/pi
        ...%i-1到i点的转角大于60度
    && sqrt(Trace_vector(1,i)^2+Trace_vector(2,i)^2)>10)%且不是偶然误差，说明第i+1个点为噪点
        DATA(:,i)=[];
        Trace_vector(:,i)=[];
        time(:,i)=[];
        err=err+1;
        continue;
    end
end
% hold on
% grid on
% ax=gca;
% ax.Color=[1,1,1];
% ax.XColor=[1,1,1].*.3;
% ax.YColor=[1,1,1].*.3;
% ax.LineWidth=1.5;
% ax.FontName='cambria';
% ax.GridLineStyle='--';
% xlabel("Longitude")
% ylabel("Latitude")
% %用于美化
% plot(DATA(8,:),DATA(9,:),'.',"Color",[0.7 0.7 0])


%曲线修复
newTrace=[];
threshold=0.001;
range=8;%拟合半径
```

```
for j=1:nowSize
    newTrace=[newTrace,[DATA(8,j);DATA(9,j)]];

    if(abs(DATA(8,j+1)-DATA(8,j))>abs(threshold))%相隔threshold度内无数据

        if(j<=range)%前range个点
            p=spline(DATA(8,1:2*range),DATA(9,1:2*range));
        elseif(j>nowSize-range)
            p=spline(DATA(8,nowSize-2*range:nowSize),DATA(9,nowSize-2*range:nowSize));
        else
            p=spline(DATA(8,j-range:j+range),DATA(9,j-range:j+range));
        end
        batchx=DATA(8,j)+threshold*sign((DATA(8,j+1)-DATA(8,j))):threshold*sign((DATA(8,j+1)-DATA(8,j))):DATA(8
        batchy=ppval(p,batchx);
        newTrace=[newTrace,[batchx;batchy]];


    end
end

% hold on
% grid on
% ax=gca;
% ax.Color=[1,1,1];
% ax.XColor=[1,1,1].*.3;
% ax.YColor=[1,1,1].*.3;
% ax.LineWidth=1.5;
% ax.FontName='cambria';
% ax.GridLineStyle='--';
% xlabel("Longitude")
% ylabel("Latitude")
% %用于美化
% plot(newTrace(1,:),newTrace(2,:),'.',"Color",[0.2 0.7 0.3]);

temp=size(newTrace);
Ptotal=temp(2);
newTrace=newTrace';
```

```
%曲线压缩:优化版道格拉斯-普克法，在用完AIS_Trace后才能使用
D=0.001;
global lost;
lost=0;
nPntSet=dp(newTrace,D,lost);
% 坐标区域修饰
hold on
grid on
ax=gca;
ax.YLim=[30.4,30.8];
ax.XLim=[114.2,114.7];
```

```
ax.DataAspectRatio=[1,1,1];
ax.Color=[1,1,1];
ax.XColor=[1,1,1].*.3;
ax.YColor=[1,1,1].*.3;
ax.LineWidth=1.5;
ax.FontName='cambria';
ax.GridLineStyle='--';
xlabel("Longitude")
ylabel("Latitude")
title('The traces of ships')
% 绘制原始数据曲线
plot(newTrace(:,1),newTrace(:,2),'Color',[0 0 1],'LineWidth',2,'Marker','*');
% 绘制新数据曲线
plot(nPntSet(:,1),nPntSet(:,2),'Color',[0 1 1],'LineWidth',2,'Marker','s');

legend('original-trace','compress-trace')


%质量评估,与上面那个不能同时开
LOSS=[];
Compression_Rate=[];
for ND=-5:0.5:0
    lost=0;
    nPntSet=dp(newTrace,10^(ND),lost);
    [m1,m2]=size(newTrace);
    [n1,n2]=size(nPntSet);
    LOSS=[LOSS,lost];
    Compression_Rate=[Compression_Rate,n1/m1]
end
figure;
[AX,H1,H2]=plotyy(0:-0.5:-5,LOSS,0:-0.5:-5,Compression_Rate,'plot');
xlabel("Log_{10} D")
grid on
legend('Compression Rate','LOSS')
set(get(AX(2),'Ylabel'),'String','Compression Rate') %左侧y轴
set(get(AX(1),'Ylabel'),'String','LOSS') %右侧y轴
set(AX(1),'XColor','k','YColor',[1 0.6 0]);
set(AX(2),'XColor','k','YColor',[0.7 0 0.6]);
title('Loss function and compression rate under different thresholds')
set(H1,'Color',[0.7 0 0.6],'LineWidth',2,'Marker','*')
set(H2,'Color',[1 0.6 0],'LineWidth',2,'Marker','*')
```

```
function nPntSet=dp(pntSet,TH,lost)%道格拉斯普克函数
% @author : slandarer
% pntSet : 二维数据点
% TH     : 距离阈值
global lost;
% 向量运算:计算所有点到首位两点连线距离
vertV=[pntSet(end,2)-pntSet(1,2),-pntSet(end,1)+pntSet(1,1)];%使下一步点乘变叉乘
```

```
baseL=abs(sum((pntSet-pntSet(1,:)).*vertV./norm(vertV),2));

if max(baseL)<TH
    % 若距离小于阈值则返回首尾点
    nPntSet=[pntSet(1,:);pntSet(end,:)];
    lost=lost+sum(abs(baseL));
else
    % 若距离大于阈值则左右两分支分别计算后拼接
    maxPos=find(baseL==max(baseL),1);
    L_PntSet=dp(pntSet(1:maxPos,:),TH,lost);
    R_PntSet=dp(pntSet(maxPos:end,:),TH,lost);
    nPntSet=[L_PntSet;R_PntSet(2:end,:)];
end
end
```

```
function Is_cross = iscro(A,B,C,D) %判断AB和CD线段是否相交,q其中ABCD均为2*1坐标

if(max(C(1),D(1))<min(A(1),B(1)) || max(C(2),D(2))<min(A(2),B(2)) ||
    max(A(1),B(1))<min(C(1),D(1)) || max(A(2),B(2))<min(C(2),D(2)))%快速排斥判断
    Is_cross=0;
else
    T1=cross([(A-D);0],[(C-D);0]);
    T2=cross([(B-D);0],[(C-D);0]);%cross: 三维向量叉积
    T3=cross([(C-A);0],[(B-A);0]);
    T4=cross([(D-A);0],[(B-A);0]);
    if(T1(3)*T2(3)<=0&&T3(3)*T4(3)<=0)
        Is_cross=1;
    else
        Is_cross=0;
    end
end
end
```