Open Journal of Transportation Technologies 交通技术, 2022, 11(2), 115-127
Published Online March 2022 in Hans. http://www.hanspub.org/journal/ojtt
https://doi.org/10.12677/ojtt.2022.112011

Hans汉斯

基于ADS-B时空数据挖掘的大终端区空域特征分析及其改善策略研究

齐　济，李亚飞*，陈智颖，马梓清，施锦涛，高海琳

中国民航大学空中交通管理学院，天津

收稿日期：2022年2月21日；录用日期：2022年3月24日；发布日期：2022年3月31日

摘　要

由于航班总量的迅速增加，国内每个空域都存在着日益严重的空域交通拥堵问题。空域热点包含大量能有效反映空中交通拥堵的信息。因此，空中交通管理领域必须研究如何从海量航线数据中定位空域热点和热门航线。本文主要阐述了目前中国国内关于航空飞行器运动轨迹聚类分析算法的主要研究状况；介绍了广播式自动相关监视系统(ADS-B, automatic dependent surveillance broadcast)；对航迹数据采用剔除、过滤和模糊等航迹预处理方法；对预处理过后的航迹数据采用基于密度的DBSCAN聚类，对特征空域采用热点密度分类，并对热点拥堵地区进行识别。通过使用BMAP API提取航空地图(Aeronautical Chart)在python平台建立华北空域航路地图；利用DBSCAN对华北空域一天当中的航迹数据进行基于密度的聚类；将聚类结果写入空域程序进行显示得出华北机场群空域热点。研究成果表明该程序可显示华北机场群终端区空域的热点拥堵区域和冷点空闲区，依据给出的空域结构和航班时刻，可给出改善大终端区运行效率的方法。

关键词

广播式自动相关监视系统，华北机场群空域热点程序，基于密度的聚类，空域繁忙区域识别优化

Spatial Characteristics Analysis and Improvement Strategy of Large Terminal Area Based on ADS-B Spatio-Temporal Data Mining

Ji Qi, Yafei Li*, Zhiying Chen, Ziqing Ma, Jintao Shi, Hailin Gao

College of Air Traffic Management, Civil Aviation University of China, Tianjin

Received: Feb. 21st, 2022; accepted: Mar. 24th, 2022; published: Mar. 31st, 2022

*通讯作者。

文章引用: 齐济, 李亚飞, 陈智颖, 马梓清, 施锦涛, 高海琳. 基于 ADS-B 时空数据挖掘的大终端区空域特征分析及其改善策略研究[J]. 交通技术, 2022, 11(2): 115-127. DOI: 10.12677/ojtt.2022.112011

Abstract

Due to the rapid increase of the total number of flights, every airspace in China has an increasingly serious problem of airspace traffic congestion. Airspace hot spots contain a lot of information that can effectively reflect air traffic congestion. Therefore, the field of air traffic management must study how to locate airspace hot spots and hot routes from massive route data. In this paper, the main research status of the cluster analysis method of aircraft trajectory in China is described. This paper introduces the automatic dependent surveillance broadcast system (ADS-B). The track data are processed by eliminating, filtering and blurring. DBSCAN clustering based on density was used for the pre-processed track data, and hotspot density classification was used for the characteristic airspace, and hotspot congestion areas were identified. Based on the Python platform, By using BMAP API to extract Aeronautical Chart, the route map of North China airspace was established on Python platform. DBSCAN is used to cluster the daily track data based on density in north China airspace. The clustering results are written into the airspace program to display the airspace hot spots of North China airport group. The results show that the program can solve the hot spot congestion area and cold spot idle area in the terminal area of North China Airport group, and provide a method to improve the operation efficiency of large terminal area according to the given airspace structure and flight time.

Keywords

Broadcast Automatic Correlation Monitoring System, Hot Spot Program of North China Airport Group, Density-Based Clustering, Optimization of Busy Area in Airspace

1. 引言

民航高速发展的今天，民用航空已经成为经济发展的主要来源之一，但同时也产生了浪费能源，航空空域资源短缺等问题，有限的空域内资源不足已经成为民航发展的主要挑战之一。对于分析航线特征的大多数航线管制类工作，目前很少有以航线的空间和时间特征为基础，并在分析的空域临界点加以应用的研究。研究如何根据航行路线的空间和时间特征发展集群并改进其处理方法，以及研究航行路线集群在空域临界点的应用势在必行。

国内外已有不少研究者对航迹聚类分析算法和航迹系统聚类方式有着较广泛的研究，徐涛[1]等人根据航迹点法向距离的航迹聚集理论进行研究工作，克服了由于飞行器速度差异造成的航迹点对选取不匹配问题；吴欣蓬[2]等人在基于密度的噪声应用中，采用了高层次分割策略，局部异常参数和快速覆盖树，提出了考虑了局部异常参数的速度、方向和高度，基于密度的聚类方法；黄天静[3]等人将多维 Hausdorff 距离应用到层次聚类算法中以替代原有的欧式距离公式；马兰[4]等人通过对 ADS-B 历史飞行数据深入挖掘，对航线、机型等数据进行统计分析、滤波降噪处理再进行 CURE 聚类分析；刘继新[5]等人采用 RDAE 降维方法提取末端区域轨道集的非线性特征，采用多种正则化方法约束内部的低维流形。选择了两种常用轨迹聚类模型：主成分分析(PCA)结合噪声空间密度聚类(DBSCAN)算法和动态时间规整(DTW)结合 DBSCAN 算法；Zeng Weili [6]等人提出了一种基于深度自编码器(DAE)和高斯混合

模型(GMM)的轨迹聚类方法来挖掘终端空域的通行交通流模式对聚类结果的直接可视化和降维可视化；Xiaver Olive [7]等人提出了一种新的方法来分离受操作程序约束的区域内的空中交通轨迹。将该技术应用于图卢兹末端操纵区域(TMA)的一组真实轨迹上；Iulian Sandu Popa [8]等人采用了路网空间下基于密度的轨道系统聚类方法 NETSCAN 开展了轨道聚类分析算法研究工作，该办法采用先根据移动对象所路过的道路信息估算出路线中比较繁忙的部分，随后再按照已确定的密度参数对子轨道段采用聚类分析法。

鉴于此，本文提出了一套基于 ADS-B 时空数据来挖掘华北大终端区空域特征的空域程序。此程序对航迹数据采用剔除、过滤和模糊等航迹的预处理方法，对预处理过后的航迹数据采用基于密度的DBSCAN聚类并对特征空域采用热点密度分类，建立了基于 python 平台使用 BMAP API 提取航空地图(Aeronautical Chart)中华北空域机场群四个机场的位置坐标以及机场附近的所有位置报告点构成华北空域程序。利用 DBSCAN 对处理过后的航迹数据进行聚类，之后将聚类结果写入空域程序进行显示得出华北机场群空域热点并在此基础上通过对空中交通拥堵等级的区分有目的地进行交通流量管控等方法以缓解空域拥堵问题。

2. 航迹数据的预处理与数据信息挖掘

ADS-B 航迹预处理

本项目的数据来源于华北机场群的 ADS-B 设备，数据包括航空器的经度、纬度、速度、高度、航向、时间等。作用半径为 180 海里，因为数据中存在不完整的航班轨迹、ADS-B 数据缺失等，所以需要对其进行预处理。以下是航空器 ADS-B 数据处理步骤：

1) 将数据轨迹按航班号进行分类

2) 重复数据点删除。处理数据中发现，航迹点的重复分为三维位置点重复以及时间重复，重复的数据会使之后计算产生偏差，所以要将重复数据找出并删除，并且要遍历全部的航迹点。

3) 剔除飞行高度始终低于一定高度之下的航班的数据。ADS-B 在低空和超低空可能存在数据不准确或缺失等情况，为了达到精确预测航迹的目的，要在处理数据前删除不准确的数据。本项目删除了全部航迹点都在 500 m 以下的航班的数据。

4) 重新生成航迹轨迹序列。因为 ADS-B 的数据轨迹点疏密不均匀(标准间隔为 1 秒一次)所以数据中常常发生重复以及间隔过大的情况，所以在数据研究之前应当将数据位置信息与速度信息相结合，重新计算生成以秒为单位的航迹序列点。

5) 重新计算 1) 中航空器的数据轨迹序列，更新存入 Excel 文档和 text 中。

因为 ADS-B 数据质量问题或存在部分数据缺失的情况，难以辨认其航班序列，需要进行以下操作来确认航班信息。

1) 相关航迹的水平速度应该大于等于 X 最小值 $V_{\min}$，且小于等于航迹 X 最大值 $V_{\max}$，公式如下：

$$v_{\min} \le \frac{|x_i - X|}{|t_i - t_x|} \le v_{\max} \tag{1}$$

2) 相关航迹的垂直速度需大于等于 X 最小值 $V_{\min}$，且小于等于航迹 X 最大值 $V_{\max}$，公式如下：

$$v_{\min l} \le \frac{|x_i - X|}{|t_i - t_x|} \le v_{\max} \tag{2}$$

除以上两个条件，另外需要航线角 α_0 满足一定条件作为判断：

$$|\alpha| \le \alpha_0$$

其中矢量 α 为 $x_i - x_{i-1}$ 和 $x_{i+1} - x_i$ 之间的夹角，为提高航迹的匹配概率，α_0 要采纳较大值。最终得到了处理后的数据。

$$\alpha = \arccos\left[\frac{(X_{i+1} - X_i)(X_i - X_{i-1})}{|X_{i+1} - X_i||X_i - X_{i-1}|}\right]$$

3. 基于三维航迹的 DBSCAN 聚类算法

3.1. 航迹格式与前期数据导入

步骤一：航迹格式构建。航迹由一系列航迹点组成，能够有效描述航空器的空间位置信息随着时间的发展所产生的变化。每个航迹点由 5 个属性构成，分别是时间，航班名称，经度坐标，纬度坐标，高度坐标。

时间：航迹点出现的时刻。

航班呼号：航班的 ID。

经度坐标：航迹点所在的经度坐标。

纬度坐标：航迹点所在的纬度坐标。

高度坐标：航迹点所在的高度坐标。

步骤二：为了把前期分类过的航迹信息导入聚类程序，需要定义一个方法 read_data 接收一个参数 data_file，之后定义二维表列名，遍历 data_file[0]后取到一个 excel 文件 fileName。

定义变量 data_temp 等于读取 fileName 取出前五列数据，将 data_temp 的列名改为 def_to_show 的列名，然后按照分钟进行正序排序。之后根据时间分组重新构建索引 data_temp_average 的 flight 列等于 data_temp.flight，将处理好的 data_temp_average 按照 data_temp 列名重新给 data_temp 赋值。将 data_temp 插入到 def_to_show 里，返回 def_to_show。

步骤三：坐标转换程序。本程序为了得到航迹间相对位置，作为衡量航迹相似性度量的依据，需要将经纬度与高度转换为同一单位制，因此需要将航迹点进行坐标转换，同时为了将聚类结果与航图相对应，得出空域热点在航图上的位置，需要将位置信息转换为米制，而我国大多采用等角正轴割圆锥投影，因此采用兰伯特投影作为本文的坐标转换依据。

利用兰伯特投影正解公式，地球参考坐标系选用 wgs84，将经纬度转换为以米为单位的三维坐标，再将高度以英尺转换为米制高度。设假定的原点的纬度为 φf，则假设的原点的经线为 λf，则假设的原点的纬度转换坐标为 NF，则假设的原点的经线转换坐标为 EF，地球第一标准纬线 $\varphi 1$，地球第二标准纬线 $\varphi 2$，纬度转换坐标为 n，经线转换坐标为 e，则地球长零点五径为 a，太阳短零点五径为 b，则地球的第一偏心率为：

$$e = \frac{\sqrt{a^2 - b^2}}{a} \tag{3}$$

兰伯特正解公式：

$$N = NF + rF - r\cos\theta \tag{4}$$

$$E = EF + r\sin\theta \tag{5}$$

$$m = \frac{\cos\varphi}{1 - e^2\sin^2\varphi} \tag{6}$$

由以上基础量可求得：

$$\theta = n(\lambda - \lambda F) \tag{7}$$

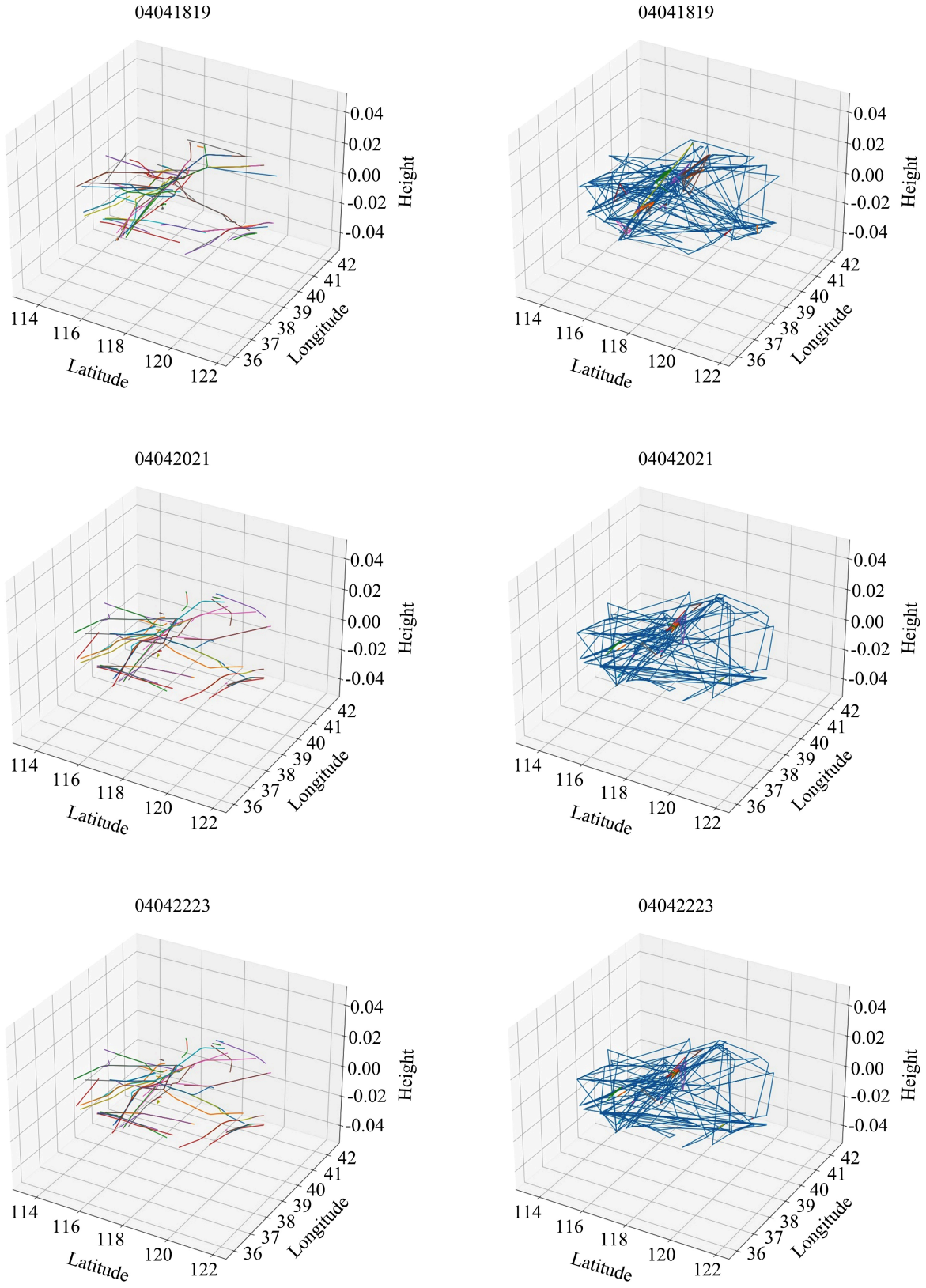

Figure 1. Original track and clustering results
图 1. 原始航迹与聚类结果

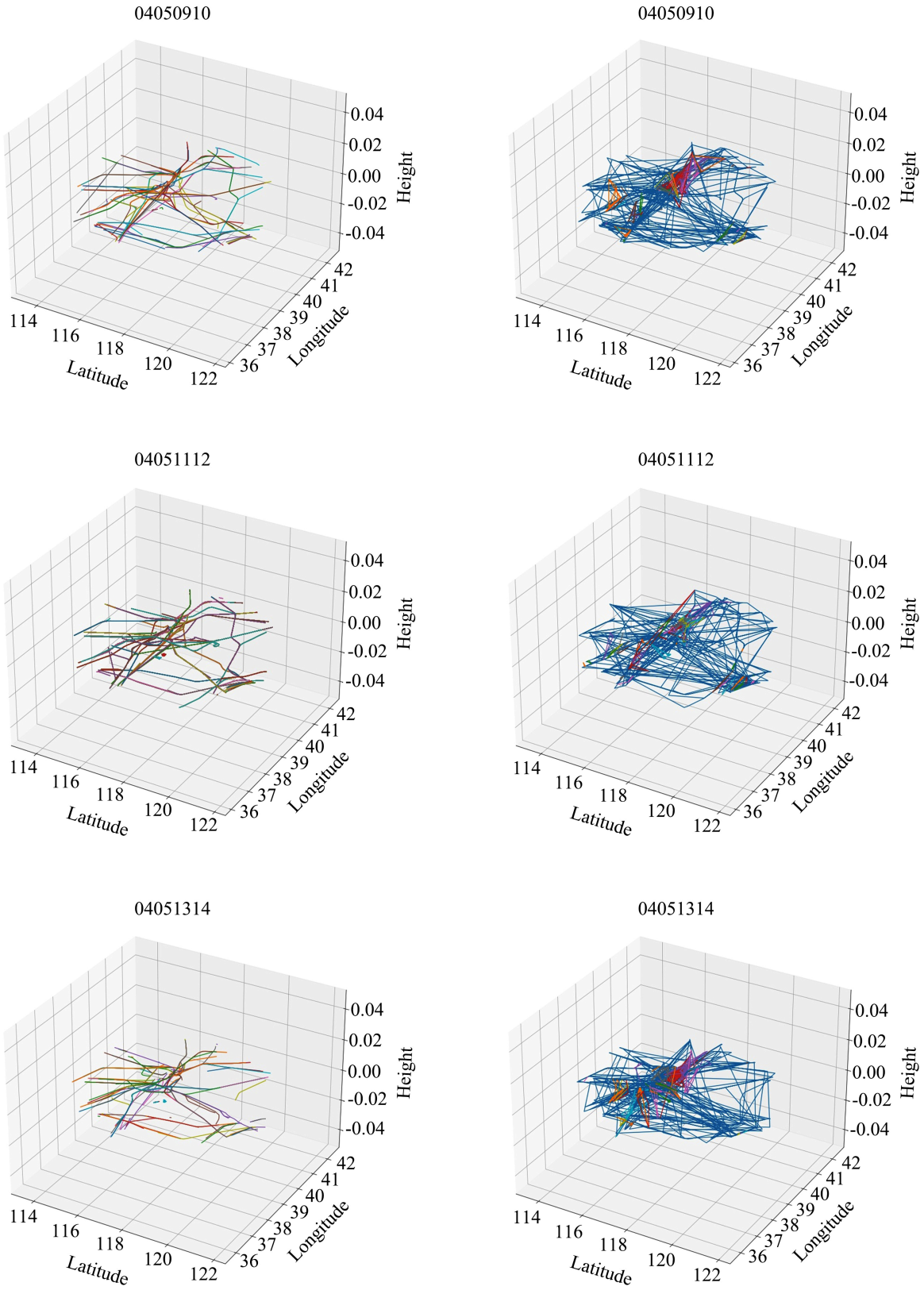

Figure 2. Original track and clustering results
图 2. 原始航迹与聚类结果

3.2. 航迹聚类程序

首先给出邻域半径：Eps 和核心目标在邻域半径内的最小点数 MinPts。本文里 Eps 取值 2500 米，Minpts 的取值根据不同空域的情况会有所变化，本程序包含庞大数据量，每个集合内航迹点所属航迹数量 tracknum 向上取整作为 MinPts，筛选出 tracknum 在 MinPts 或 MinPts 之上的中心航迹点作为核心航迹点。从任意一点 p 出发，将其标识为 visited"，并检测其是否为核心航迹点(即 p 的 Eps 邻域最少有 MinPts 个对象)，若非核心航迹点，则将其标识为噪声点。否则就给 p 建立了一个新的子群 C，同时将 p 的 Eps 邻域中的每个对象都放在候选集 N 中。迭代的将 N 中所有不包括其它簇的对象都加至 C 中，在此步骤中，首先针对 N 中标记为 unvisited 的对象 p，把它标识为 visited"，并且检查它的 Eps 邻域，假定 p 也为核心航迹点，则 p 的所有 Eps 邻域中的对象都被加至 N 中。接着增加对象至 C 中，直至 C 无法扩充，或直到 N 为真空。此时，簇 C 才完全生成。在所有余下的对象中随机选取下一个未访航迹点，并重复以上的所有步骤，直至全部航迹点对象均被访问。直至簇的数量不再改变，此时得到最终的聚类结果即处于热点空域的航迹点。航迹聚类核心代码见附件。

以下为四月四日晚十八点至二十三点，四月五日九点至十四点的原始航迹和聚类结果(见图 1、图 2)：

其中左边是每两个小时的原始航迹，为了立体显示，将图像在三维坐标系以二维形式显示；右边是对原始航迹一一对应的基于密度的聚类结果。其中不是蓝色的表示为未达到聚类标准的航迹，以上结果说明聚类航迹所处空域的单位时间(10 分钟)内的流量大于全部航迹所处空域的单位时间(10 分钟)内的流量平均值，同时说明聚类航迹所处航路为该空域主要使用的航路。

4. 基于 BMAP API 的空域特征程序

本文中使用的华北机场空域地图程序包含 ZBAA、ZBAD、ZBTJ、ZBSJ 四个机场的位置坐标以及机场附近的所有位置报告点，再将前文已聚类过的航迹数据通过 python 平台写入地图程序，加以各种要素即构成空域特征模型，本部分介绍空域特征模型的制作方式。

4.1. 地图选择

目前，百度有限公司的 BMap API 和谷歌有限公司的 Google.Maps API 代表了 Map API 发展的两个主流方向，从开发与维护的不同视角出发，两者都能够解决大部分应用需要。从地图范围上来看 BMap 只包含了全国范围内，而 Google.maps 则提供了世界范围内的地图；从精确性上来看，BMap 提供了小数点后六位数的精确度，如：中国民航大学(117.360129，39.114280)。该系统主要收录中国国内部分民航机场的地理坐标数据，与数据安全、地理信息安全相关，若直接调用 Google API 接口，可能会造成泄密。谷歌地图在我国大陆尚未取得运营牌照，和国内外主要浏览器的相容性较差，地图访问服务可能无法使用。而 Python 中的地图可视化库很多，多用于制作静态地图，而制作交互式地图可选择 pyecharts、folium。Pyecharts 库中又包含 Map、Geo 和 BMap 三类。综上，本文将采用更加安全稳健的 BMAP API 作为系统开发接口。BMAP API 基于百度地图，并可通过 JavaScript 直接纳入网页。API 中使用了大量工具来管理地图，并利用各种服务把内容加载到地图，从而使得使用者可以在自己的网页上制作强大的地图应用程序[9]。

4.2. 地图设计

首先在系统上安装 pyecharts 库，以及 HTML5 的库。HTML 语言的全称为(Hyper Text Markup Language)即超文本标记语，也是目前网络上使用得最为普遍的文字标记语。HTML5 普遍适用于现有的主流浏览器，如 Edge (Internet Explorer)、Chorme、Firefox、Safari 等。使用了最新的 HTML 的相关标签技术和规范，

能够全面支撑移动设备如手机、平板等的浏览功能，均可以通过自带浏览器实现地图的显示，界面也能够针对用户设备，实现自适应变化。并且，百度地图还需要一些 HTML 元素作为容器以展示在网页上，设置地图初始化配置项，创建 1600*800 像素的图表画布。申请成为百度开发者，获得使用地图 API 接口的权限，获取 AK 码写入代码中。根据小组成员提供的航迹地理位置数据，使用 Excel 批量转换成适用于 BMap 的经纬度信息。航路点和航迹部分，由于数据量较小，将具体经纬度直接写入代码。参照 pycharts 官网 BMap 示例对地图进行初始化，使用 add_coordinate 批量添加航路点坐标。随后，对航迹、航路点进行数据可视化。

八小时时段内，数据量高达百万，采用 JSON (Java Script Object Notation)格式新增多个坐标点。JSON，即 JavaScript 对象表示方式，是一个轻量级的数据格式，是保存和转换文件信息的标准语言。JSON 中数据类型的标准书写格式为：名称/值对，并行的数据类型之间用“，”分隔，映射用“：”表示，并行数据的集合用“[]”表示，映射的组合则用“{}”表示[10]。4 月 3 日八小时时段内的 JSON 数据表示为：{编号 1"：[经度，纬度]，编号 2"：[经度，纬度]}。调用 zip 函数，接受一系列可迭代对象作为参数，将不同对象中相对应的元素(即航班与对应经纬度坐标)打包成一个元组，返回由这些元组组成的 list，由此读入百万数据，并在图上作出相应坐标。再添加华北四个机场经纬度坐标，在地图上标示出来。最后对全局配置项进行调整，完成对基础地图设计。地图程序设计部分核心代码见附件。

4.3. 空域热点识别

DBSCAN 算法的基本原理是：对聚类中的所有数据对象，在指定的半径范围邻域内的统计对象必须多于一个阀位。其计算方式简单，对噪声点也不敏感，并且能够发现任何形状的簇，但是仍然存在缺点：由于必须在整个数据空间建立树，算法中需要较大的 IO 开销。对于本程序来说，DBSCAN 算法对于输入参数缺乏很全面的科学规范来作依据，这将导致人为影响的原因变得很多，参数选择略有误差对于聚类的效果将产生完全不同的效应。所以为了能将热点识别结果便捷写入程序中，需要使得各个参数的阈值易于调整，本程序利用 HDBSCAN 方法的 probability 属性创造了易于调整的阈值来控制拥堵程度的变量。

参数选择：

min_cluster_size：一个类中至少要有 min_cluster_size 个样本，这个参数越大，最终的聚类种类数会越少。本程序中将此参数定为 300。

min_samples：一个点邻域范围内至少有 min_samples 个样本，才会被视为核心点；提供的 min_samples 的值越大，聚类越保守，将更多的点声明为噪声，并且聚类将被限制在逐渐密集的区域。本程序中将此参数定为 1000。

由于 HDBSCAN 与 DBSCAN 核心思想相近，本节不做过多介绍，仅介绍本程序核心代码中 HDBSCAN 的具体用法。本地图使用的是 HDBSCAN 聚类算法的 probability 项对每组点进行区分，prob 越大代表属于该类的可能性越大，相对来说比较拥堵，根据拥堵的程度划分了四个等级，分别为严重拥堵(prob >= 0.75)，中度拥堵(0.5 <= prob < 0.75)，一般拥堵(0.25 <= prob < 0.5)，不拥堵(prob < 0.25)。对标签为−1 的点直接设置为离散。

4.4. 基于 BMAP API 的空域特征程序解析结果

本程序使用颜色区分空域繁忙程度，红色代表严重拥挤，橙色中度拥挤，黄色一般拥挤，绿色则是聚类结果中未达到拥挤聚类指标标准的。下图 3 为选取了具有代表性的华北空域机场群 2020 年四月四日 18 点至四月五日 16 点的空域特征模型。

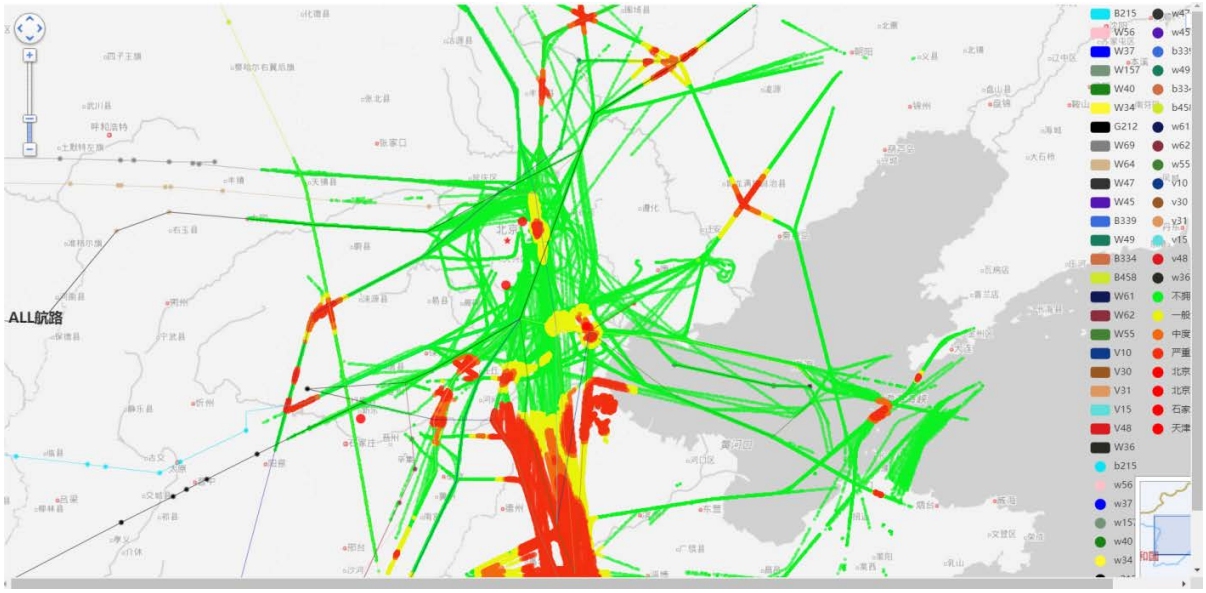

Figure 3. Spatial model overview
图 3. 空域模型总览

下面是空域热点程序分析结果及优化意见：B339 航线 HUAIROU 报告点至 SOTMU 报告点中度繁忙。北京首都机场 IDKEX 系列，DOTRA 系列，BOTPU 系列，IGMOR 系列进离场程序繁忙，其中 LULTA 程序 AA511~AA513 段严重繁忙，建议把流量分配给 MUGLO，ELKUR 进离场程序。

天津滨海机场向南方向的 GUVBA，DUMAP 程序繁忙，西南角航线量少，可选择从 ELKUR 程序或从右侧选择 MUGLO，IGMOR 程序离场(见图 4)。

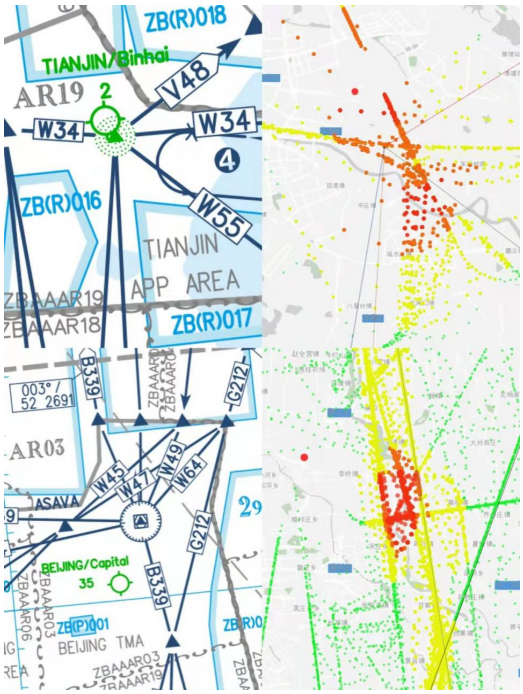

Figure 4. Beijing Capital Airport and Tianjin Binhai Airport
图 4. 北京首都机场和天津滨海机场

W157 航线 AVBOX 报告点显著繁忙，应减少从 ZBAD，ZBTJ 等机场分流来的航班，应分流至 OMDEK，ELKUR 等比较畅通的航路点。W157 京津地区至济南进近区间的航段非常繁忙。

W56 航线 NUDKU 至 AVLUV 航段拥堵，可分流至与其航线方向大致相同的 W37 航线。AVLUV 至 GUVRI 客流量大，有多条航线例如 W85 交汇，GUVRI 作为附近几条航线的枢纽点，DUGEB 为京津地图机场群进近区域边界上的报告点，汇合的航班密度也偏大，相对来说河北石家庄正定机场在 4 月 5 日的客流量不大，程序上并未显示有显著拥堵区域(见图 5)。

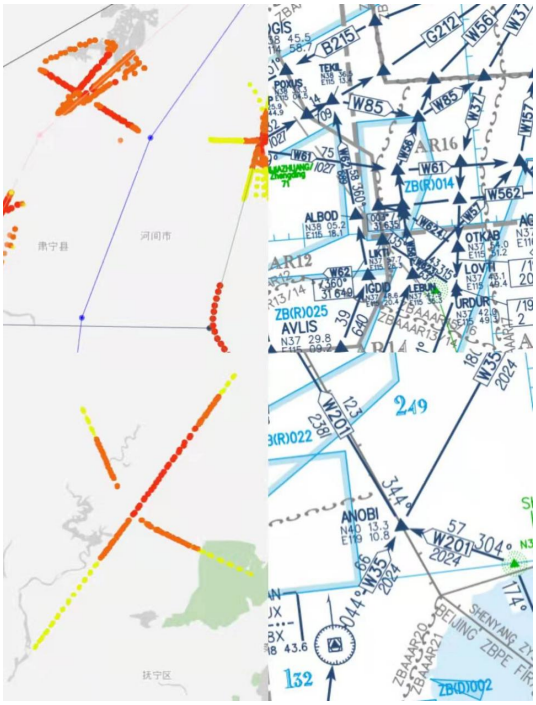

Figure 5. Regional air route
图 5. 地区航路

东北方秦皇岛市上空 W35 与 W201 交汇处 ANOBI 报告点航线繁忙。其上方省份以及济南下方地区并不在研究区域范围以内，但程序显示贯穿南北的 W56，W37，A461，A593 等航线客流量比起各机场附近显著增大，这是由于主干航线会并入各个机场来的所有航班，本研究仅基于华北机场群空域，故不对主干航线的航路繁忙问题进行更深一步的探讨。

5. 结束语

本文是以航线的空间和时间特征为基础，并把分析的空域热点临界点加以应用的研究。通过研究根据航行路线的空间和时间特征发展集群并改进其处理方法，以及研究航行路线集群在空域临界点的应用是本空域程序的核心用途。使用以密度进行聚类处理的航空器飞行信息数据(高度、经度、纬度、速度)写入空域特征程序，基于分级的方式确定区域热点参照航图中的各种信息来确定实时空域情况。该程序优点在于有一套从 ADS-B 数据到可视数据的完整程序，只需初始数据即可得到所需空域的热点情况，并且经过预处理和聚类的数据精度高、可用性高。算法的升级空间很大，程序如果得到相应的机会，数据就可以得到进一步的拓展。实验表明，从起始数据到最终形成热点航路，程序复杂度较高，所用时间较长，适合面向数据规模较小且对聚类精确度要求较高的应用场景，下一步的工作重点是对程序算法的进

一步合并和简化，以争取更短的运算时间。

基金项目

中国民航大学大学生创新创业训练计划项目资助(项目编号：202110059088)；教育部人文社科基金项目资助(21YJCZH075)。

参考文献

[1] 徐涛, 李永祥, 吕宗平. 基于航迹点法向距离的航迹聚类研究[J]. 系统工程与电子技术, 2015, 37(9): 2198-2204.

[2] 吴欣蓬, 汤新民, 毛继志, 郭鸿滨, 管祥民. 基于密度聚类与匹配算法的异常飞行行为挖掘[J]. 南京航空航天大学学报, 2021, 53(6): 863-871.

[3] 黄天镜. 基于多维特征的 ADS-B 数据异常检测方法研究[D]: [硕士学位论文]. 天津: 中国民航大学, 2020.

[4] 马兰, 高永胜. 基于 ADS-B 数据挖掘的 4D 航迹预测方法[J]. 中国民航大学学报, 2019, 37(4): 1-4.

[5] 刘继新, 董欣放, 徐晨, 杨光, 江灏. 基于密度峰值的终端区航迹聚类与异常识别[J]. 交通运输工程学报, 2021, 21(5): 214-226.

[6] Zeng, W.L., Xu, Z.F., Cai Z.P., Chu, X. and Lu, X.B. (2021) Aircraft Trajectory Clustering in Terminal Airspace Based on Deep Autoencoder and Gaussian Mixture Model. *Aerospace*, **8**, Article No. 266. https://doi.org/10.3390/aerospace8090266

[7] Olive, X. and Morio, J. (2019) Trajectory Clustering of Air Traffic Flows around Airports. *Aerospace Science and Technology*, **84**, 776-781. https://doi.org/10.1016/j.ast.2018.11.031

[8] Popa, I.S., Zeitouni, K., *et al*. (2015) Spatio-Temporal Compression of Trajectories in Road Networks. *GeoInformatica*, **19**, No. 1. https://doi.org/10.1007/s10707-014-0208-4

[9] 朱剑明, 余朋. 基于百度地图 API 的全国机场天气警报系统设计与实现[J]. 气象科技, 2017, 45(2): 247-253

[10] Chaffer, J. jQuery 基础教程[M]. 北京: 人民邮电出版社, 2008.

附件

1. 经纬度坐标转换程序核心代码

```
def transformlat(lng, lat):

ret = -100.0 + 2.0 * lng + 3.0 * lat + 0.2 * lat * lat + 0.1 * lng * lat + 0.2 * math.sqrt(math.fabs(lng))
ret += (20.0 * math.sin(6.0 * lng * math.pi) + 20.0 *math.sin(2.0 * lng * math.pi)) * 2.0 / 3.0
ret += (20.0 * math.sin(lat * math.pi) + 40.0 *math.sin(lat / 3.0 * math.pi)) * 2.0 / 3.0
ret += (160.0 * math.sin(lat / 12.0 * math.pi) + 320 *math.sin(lat * math.pi / 30.0)) * 2.0 / 3.0
return ret
```

2. 航迹聚类核心代码

```
lng_list,lat_list=[],[]
for i in range(X.shape[0]):
lng, lat=X.iloc[i,2]
lng, lat=transformlat(lng,lat),transformlng(lng,lat)
lng_list.append(lng*1000)
lat_list.append(lat*1000)X['lng_trans']=lng_list
X['lat_trans']=lat_list
X_for_model=X[["hight","lng_trans","lat_trans"]]
db=DBSCAN(eps=2500,min_samples=50).fit(X_for_model)
labels=db.labels_
```

3. 程序设计部分核心代码

```
def panduan(x):
    if x<=0.75:
        res=0
    elseif x<=0.95:
        res=1
    elseif x<1:
        res=2
    else:
        res=3
    return res
df_all['level']=df_all.prob.apply(lambda x: panduan(x))

df_busy=df_all[df_all.level==3]
df_medium=df_all[df_all.level==2]
df_less=df_all[df_all.level==1]
df_fair=df_all[df_all.level==0]
```

```
df_fair=pd.concat([df_fair,df_all_buyongdu])

long_busy,lat_busy=df_busy.long, df_busy.lat
long_medium,lat_medium=df_medium.long, df_medium.lat
long_less,lat_less=df_less.long, df_less.lat
long_fair, lat_fair=df_fair.long,df_fair.lat

long_lat_busy=[z for z in zip(long_busy,lat_busy)]
long_lat_medium=[z for z in zip(long_medium,lat_medium)]
long_lat_less=[z for z in zip(long_less,lat_less)]
long_lat_fair=[z for z in zip(long_fair,lat_fair)]

place_busy=df_busy.place.tolist()
place_medium=df_medium.place.tolist()
place_less=df_less.place.tolist()
place_fair=df_fair.place.tolist()

for i in range(len(place)):
    b1.add_coordinate(str(place[i]),long[i],lat[i])
```