北京交通大学

硕士专业学位论文

基于浮动车轨迹数据的路径规划研究

Research on Path Planning Based on Floating Car Data

作者：赵风萍

导师：李浥东

北京交通大学

2020 年 6 月

学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。学校可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：赵凤萍　　导师签名：

签字日期：2020 年 6 月 15 日　　签字日期：2020 年 6 月 15 日

学校代码：10004 密级：公开

北京交通大学

硕士专业学位论文

基于浮动车轨迹数据的路径规划研究

Research on Path Planning Based on Floating Car Data

作者姓名：赵风萍 学　　号：17127149

导师姓名：李浥东 职　　称：教授

工程硕士专业领域：软件工程 学位级别：硕士

北京交通大学

2020 年 6 月

致谢

时间飞逝，回首再次踏入校园时的欣喜和憧憬，转眼间研究生生涯即将结束。借此机会，向所有给予我支持、帮助的老师、同学以及朋友表示衷心的感谢。

首先要感谢我的导师李浥东老师，研究生期间李老师渊博的知识、独到的见解、严谨的态度对我影响很深。从论文开题、查阅相关论文、确定研究方向，到论文初稿完成，李老师提出了很多宝贵意见。此外还要感谢赵宏伟老师、章子凯师兄、万群、林雅婷、党正午、田金秀、周佳棋在我研究生学习、生活、论文完成过程中给与的帮助和关心。深深感谢我的父母家人对我学习的默默支持和精神上的鼓励，是你们给了我继续攻读研究生的勇气。

特别感谢所有支持过我、帮助过我、批评过我、鼓励过我和理解过我的人们，祝他们工作/学习顺利、身体健康！

摘要

移动设备的广泛应用，使得人们出行越来越依赖于基于位置服务的路径规划。随着城市化进程的加快，汽车普及程度提高，交通拥堵问题日渐严重，合理的路径规划可以改善人们的出行体验，降低出行成本。

传统的路径规划一般是基于计算的理论上的最优路径，但未考虑人们的经验习惯及环境因素。浮动车轨迹反映城市道路的行驶规律，也是司机的驾驶经验智慧的体现，综合考虑了道路等级、红绿灯数量、拥堵程度及周围环境等各种因素。通过对浮动车数据进行挖掘，实现更优的路径规划是本文的主要任务。

本文主要从浮动车轨迹数据本身出发，以数据为驱动，对轨迹特征进行分析，提取出最大程度上包含司机典型经验的完整路线，实现基于经验路线的分层路径规划。本文的主要研究内容：

（1）浮动车轨迹数据的经验路线提取。针对轨迹聚类，本文提出一种轨迹间距离度量的方法并根据轨迹间相似度进行聚类提取质心经验轨迹。然后对质心经验轨迹进行地图匹配，生成经验路线。

（2）使用改进的 HMM 算法进行地图匹配。针对地图匹配的准确性和投影情况的多变性，提出改善的观测概率计算方法和转移概率计算方法，来降低轨迹点的错误投影概率。针对地图匹配的计算效率，提出网格化的存储和基于网格范围的计算来提高观测概率的计算效率，并通过计算并缓存地图上任意两点的最短距离，大幅度减少状态转移概率的计算量。

（3）基于经验路线的路径规划研究。针对路径规划方法的研究，提出基于经验路线的分层路径规划方法。通过实验对比，基于经验路线的路径规划结果从时间和距离上呈现一定优势。

关键词：路径规划；浮动车经验；轨迹聚类；HMM 算法；地图匹配

ABSTRACT

The widespread use of mobile devices has made people more and more dependent on path planning based on location services. With the acceleration of urbanization, the popularity of automobiles has increased, and the problem of traffic congestion has become increasingly serious. Reasonable path planning can improve people's travel experience and reduce travel costs.

Traditional path planning is generally based on the theoretical optimal path of calculation, but it does not take people's experience, habits and environmental factors into account. The floating car trajectory reflects the driving laws of urban roads, and also reflects the wisdom of drivers 'driving experience. It comprehensively considers various factors such as road grade, number of traffic lights, congestion level and surrounding environment. It is the main task of this paper to mine floating car data to achieve better path planning.

This article mainly starts from the floating car trajectory data itself, based on the data, analyzes and mines the hidden traffic information, and extracts the complete route that contains the driver's typical experience to the greatest extent, to realize the hierarchical route planning based on the experience route. The main research content of this article:

(1) Extracting the empirical route of floating car trajectory data. For trajectory clustering, this paper proposes a method of distance measurement between trajectories and extracts centroid empirical trajectories based on clustering of similarity between trajectories. Then, map the centroid experience track to generate an experience route.

(2) Use the improved HMM algorithm for map matching. Aiming at the accuracy of map matching and the variability of projection conditions, improved observation probability calculation methods and transition probability calculation methods are proposed to reduce the erroneous projection probability of trajectory points. In view of the calculation efficiency of map matching, grid storage and grid-based calculation are proposed to improve the calculation efficiency of observation probability. By calculating and buffering the shortest distance of any two points on the map, the calculation of state transition probability is greatly reduced the amount.

(3) Research on path planning based on empirical routes. Aiming at the study of path planning methods, a hierarchical path planning method based on empirical route

layers is proposed. Through experimental comparison, the results of path planning based on empirical routes present certain advantages in terms of time and distance.

KEYWORDS：Path planning; Floating Car experience; Trajectory clustering; HMM algorithm; Map matching

目录

1 引言

1.1 研究背景及意义

移动互联网的发展、智能手机设备的普及、基于位置服务的广泛应用，为人们生活带来很大便利。在规划出行上，使得人们摒弃了传统的纸质地图，更加依赖智能设备的路径规划服务。据统计 2018 年我国手机地图用户超过 7.2 亿人，这个数据表明了路径规划的巨大用户需求。另一方面，随着我国经济快速增长和城市化进程加快，城市人口和车辆不断增加，同时城市道路也在飞速增长，路网规模庞大且复杂。复杂多变的交通状况下，城市交通拥堵、交通事故频发等导致道路与车辆矛盾日益恶化，如何规划合理的行驶路径，改善人们的出行体验，降低出行成本，成为目前城市交通亟待解决的问题。

空间定位技术、信息通信技术和移动终端技术日趋成熟，基于位置的服务（Location Based Services，LBS）、移动社交网络、物联网和智慧城市等广泛应用，产生了大量蕴含时空信息的轨迹数据。随着交通数据采集技术的不断发展，及时获取路网中实时的交通轨迹数据已成为可能，浮动车是收集交通数据最有效的方式之一，具有覆盖范围广，易采集等特点。浮动车所记录的行驶过程中的位置，方向和速度等信息称为浮动车轨迹数据（Floating Car Data，FCD）。基于 FCD 采用各种数据挖掘、图形识别、智能算法等获取价值信息的知识发现过程被称为浮动车技术（Floating Car Technology，FCT）[1]。浮动车数据蕴含了丰富的时空特性，反映了人们的出行规律和出行特征，同时也反映了实地的路网信息和交通状态信息。例如，浮动车轨迹的速度信息和轨迹点密度能反映当前道路的拥堵情况；轨迹行为可以反映路网信息等。如何对大规模轨迹数据进行挖掘，发现其中蕴含的更多有价值的信息，为人们生活提供便利，成为目前学者们在数据挖掘领域的一个重要研究方向。通过对轨迹数据进行有效的挖掘，能够为交通路线优化、个性化推荐路线、交通信息预测、城市规划等提供有效的解决方案。

目前装载 GPS 浮动车主要是方便管理和安全的公共交通工具公交车或者出租车以及出行平台的车辆等，这些浮动车的司机有着丰富的行驶经验，尤其是出租车司机，他们是对城市路况信息最为熟悉的群体。通常情况下出租车之间也保持着信息的沟通，可以更加快速的了解实时交通信息，他们根据对不同时间段交通信息的了解，能有效躲避拥堵，选择更加合适的路径。所以一定程度上出租车的载人轨迹反映城市道路的行驶规律，也是司机的驾驶经验智慧的体现，综合考虑

了道路等级、红绿灯数量、拥堵程度及周围环境等各种因素。传统的路径规划一般是基于计算的理论上的最优路径，但未考虑人们的经验和习惯以及一些环境因素。通过对出租车的轨迹进行挖掘，提取老司机的经验路线，将驾驶者的经验智慧融入到道路路径规划中，建立合适的路径规划模型，对车辆出行具有重要的指导意义。

随着现代科技的发展，将计算机技术应用于智能交通领域，建立交通管理和交通信息服务系统，高效地服务于城市道路交通系统。浮动车载客数据进行挖掘分析，了解城市客流分布、交通车流密度，能够对城市规划、路网建设等提供依据。经验路线的挖掘，可以为驾驶者提供最具经验价值的导航信息，使人人都成为老司机，找到更好、更快、更短的行车路线，缩短路径消耗时间，降低交通成本，在一定程度上减缓城市交通运输压力，改善城市交通状况。

1.2 国内外研究现状

1.2.1 浮动车技术研究现状

FCD 技术应用发展于 20 世纪 90 年代早期，但由于移动定位的成本问题，在早期并没有大量采用，直到 2000 年美国取消了降低 GPS 精度的 SA（Selective Availability）技术，大部分车辆都装备了低成本的 GPS，使得 FCD 用于交通信息采集有了可能性[2]。

浮动车作为新型的交通信息采集技术，利用车载 GPS 定位装置在其行驶过程中定期获得车辆的位置、速度、方向等状态信息，并将这些信息数据通过无线通信技术传输到信息处理中心，经过综合处理、挖掘，可以获得城市中动态的交通数据信息。浮动车数据挖掘的基础是轨迹与地图路网数据的匹配，配准是所有基于 FCD 挖掘应用的基础，匹配的准确性和实时性直接影响挖掘有价值信息的准确性和实时性[1]。为了提高地图匹配的精度，国内外学者做了大量的研究。Bernstein and Kornhauser[3]、White[4]、Phuyal[5]等学者从几何关系方面分别基于点到点、点到直线、线与线提出不同的地图匹配算法，求出 GPS 点的匹配定位信息。Greenfeld 从拓扑关系方面提出的一种加权拓扑信息地图匹配算法[6]。Ohieng 从概率方面提出一种增强概率的地图匹配算法[7]。之后 Paul Newson and John Krumm 首先提出基于隐马尔可夫模型（Hidden Markov Model，HMM）的地图匹配算法[8]，有效提高了地图匹配的精度。因现实存在很多复杂的路网情况，地图匹配的研究也是学者们持之以恒追求的课题。

浮动车数据与道路匹配为浮动车技术在智能交通领域的探索在技术上奠定了基础。数据源方面，在我国 2014-2015 年，滴滴、uber、共享单车等新型的出行方式不仅改变了人们传统的出行，也为轨迹数据挖掘提供了精度更高、覆盖范围更广、更具代表性的数据源[9]。基于轨迹数据的挖掘在智能交通方向的应用主要有交通信息的挖掘的和地图路网数据的挖掘两个方面。交通信息的挖掘主要有行程时间估计[10]、拥堵态势分析与预测[11]、车流量分析及预测[12]、交通事件监测[13]、道路通行能力估计、路口评价等。另一方面浮动车的行车轨迹也勾勒了真实的路网结构，基于轨迹数据挖掘缺失或者新增道路，以补充完善道路网数据。Agamennoni G[14] 等通过对大规模出租车轨迹数据进行聚类，提取轨迹中心线来对道路数据进行补充。Chen C[15] 采用栅格的方式，将轨迹数据栅格化后转换成二值图像，然后对轨迹栅格数据进行提取生成道路中心线。通过挖掘路网信息可以检测道路网络的变化，为导航地图生产商提供有效的道路情报信息，针对变化信息进行地图数据的采集制作，进而实现电子地图的快速更新。

除基于地理位置的语义挖掘外，浮动车技术在用户行为语义分析方面也做了大量研究。通勤分析、潮汐现象分析、职住区域分析、出租车需求预测[16]等，基于空间地理信息系统的视觉挖掘驾驶员居住地和作息时间规律[17]。

1.2.2 路径规划研究现状

传统的路径规划主要是基于最短路径的计算，最短路径的求解是一个比较热门的图论搜索算法，最早的论文可以追溯到七十多年前甚至更早，但近几年来这个研究的热度一直高居不下。最短路径中的短包含很多维度，不仅仅是指地图经过的道路距离最短，还包括时间最短、收费距离最短、红绿灯最少等多种选择，用户可以根据自己的偏好，选择自己需求的路径。对现有的路径规划计算方法进行总结归类，主要有三种类型：静态路径规划方法、动态路径规划方法、基于分层的路径规划方法。

静态路径规划方法是在静态的电子地图道路数据上计算起点和目的地之间的最优路径，可以根据道路属性或等级的不同设置不同的权值，使最终计算出来的通行路线最优。最常见的算法有 Dijkstra 算法[18]、Floyd 算法[19]、A*算法[20]等。Dijkstra 是典型的单源最短路径，从起始点开始向外扩张计算到所有结点的最短路径，时间复杂度为 $O(n^2)$。Floyd 算法算法通过两个二维数组分别代表所有顶点到所有顶点的最短路径权值和以及对应顶点的最小路径的前驱顶点，可以处理两点间距离是负值的情况，该算法复杂度是 $O(n^3)$；A*算法是一种使用启发函数的直接

搜索方法，启发函数的距离估值与实际情况越接近，节点展开的范围就越小，搜索速度越快，A*算法没有明确的时间复杂度。除此以外，还有深度或广度优先搜索算法、Bellman-Ford 算法、SPFA（Shortest Path Faster Algorithm）算法等。

在处理庞大的路网数据时，经典算法难以满足计算时间需求，为减小图的搜索空间，基于分层路网的路径规划方法被提出。通过对基础路网数据进行路网分层、道路分区以及建立以区域为单位的路网层次拓扑关系，再进行分层搜索，大幅度提高了路径规划效率[21]。2000 年-2010 年日系车载导航仪的 KIWI 规格地图数据中路径计算数据就是采用了垂直分层、水平分块的存储方式。如图 1-1 所示，根据道路的路径通行等级进行预处理，生成 3 层路网数据，即广域层数据、基本层数据、详细层数据，每层都被划分成不同大小的块。详细层数据包含所有车辆可通行的道路数据，基本层数据则只收录县道、国省道、高速等通行等级较高的道路，广域层只收录国道高速等高等级道路。用户设置起点目的地后，车载导航仪只需要根据起点目的地所在的范围取相应的预处理好的数据块，并通过上下层间的拓扑关系进行关联进行简单计算，计算过程不需要遍历全国数据，大大节省了路径规划的计算时间。

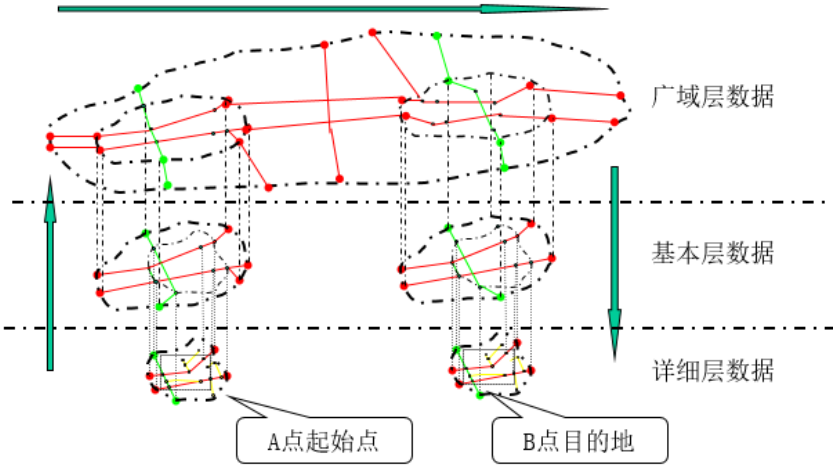


图 1-1 分层的路径计算过程

Figure 1-1 Hierarchical path calculation

城市道路交通状况在时间上存在较大的变化，车流量、交通故障等因素对路径规划有很大影响，静态路径规划难以实现智能交通系统（Intelligent Traffic System，ITS）中车辆导航路径规划。动态路径规划方法最早是由 Cooke and Halsey[22]提出的，路径计算时先把道路数据用离散化时间差的倍数表示，然后求解起点到目的之间的最优路径。地图商通过在关键点位部署车辆经过情况的感应线圈或摄像头、电子地图用户的位置移动信息、浮动车实时回传的通行信息等方式，获取道路的拥堵程度；通过用户上报、新闻发布或其他途径获得施工、临时交通管制

等信息。动态路径规划加入实时的交通信息对预先规划好的最优行车路线进行适时的调整直至到达目的地，最终得到最优路径，这也是现在智能交通的重要研究内容。孙海鹏[23] 等将实时交通信息转化为路段上的动态时间特征数据，进行路径规划。樊月珍[24] 通过预测交叉口的车流量，将交通荷载作为变量动态计算道路权重，实现路径的优化选择。

然而基于实时交通信息的动态导航，目前存在交通信息获取的可靠性和准确性问题及信息的高效数据组织问题，技术不够成熟和完善，短时交通预测不够准确。传统的最短路径算法在理论上可以得到最优路径。实际上，驾驶员更愿意根据自己的经验选择更有效的路径，而不是选择理论上的最短路径。利用出租车司机行车路线的选择经验辅助路径导航，能够很好地解决基于实时交通信息动态导航的不足。Zhuang L.等[25]利用浮动车的数据，探索驾驶员的路径选择经验，建立经验路径数据库并修复异常和不完整路径，得到备选经验路径数据集。Li 等[26]根据驾驶员的经验提取频繁的路线，但它只针对一些频繁的出发地和目的地。唐炉亮、常晓猛、李清泉[27]对出租车司机选择行驶道路的规律进行研究，建立司机道路寻径的经验知识模型，提出基于经验知识的路径规划算法。戚欣、梁伟涛、马勇[28]通过轨迹与地图数据匹配，根据访问频率选择热点路段，并对热点路段间的行驶轨迹进行聚类，生成热点路段图，最后使用改进的 A*算法实现路径规划。

目前对于浮动车驾驶经验的研究，大多数集中于交通流、路口信息等单一交通信息的研究，且并没有将挖掘的信息融入到路径规划研究中。基于浮动车轨迹数据的路径规划研究偏向于热点路段，只考虑了路段的行驶热度，不能很好的体现司机典型经验路径的完整性[29]。

1.2.3 本文研究内容

基于目前研究存在的问题，本文主要从浮动车轨迹数据本身出发，以数据为驱动，分析挖掘隐含的交通信息，并提取出最大程度上包含司机典型经验的完整路径，实现基于经验路线的路径规划。本文主要研究内容：

（1）轨迹特征分析，对轨迹的时空特征进行分析了解人们的出行规律及出行特征；轨迹行为特征分析，可以识别实地的路网信息和交通状态信息。提出基于轨迹行为特征提取的模式匹配方法进行交通信息识别，通过选择轨迹特征，训练分类器，将轨迹按行为特征进行分类，同时定义可以表述不同交通信息的各种行为模式，将分类好的轨迹行为进行交通模式匹配，进而识别当地的交通信息。出行特征和交通特征信息的识别可以为路径规划提供参考信息。

（2）经验路线提取，通过对轨迹进行聚类，有效识别选择最多的线路，并过滤掉选择偏少的分支轨迹和有绕路的噪声轨迹，提出一种轨迹间距离度量的方法，根据轨迹间相似度聚类提取质心轨迹。然后对质心轨迹进行地图匹配，生成经验路线。

（3）使用改进的 HMM 算法进行地图匹配，算法创新点有：

1）对轨迹点和地图数据进行网格化存储，增加索引，提高计算效率；计算观测概率时根据网格限定计算范围，减少计算量。

2）计算地图上任意两个 node 点的最短路径距离进行缓存，大幅减少状态转移概率的计算时间。

3）设置两个系数，增加投影的准确性，计算观测概率部分增加轨迹点方向和道路方向差异量的系数θ，差异量越小，观测概率越大；计算转移概率时，当相邻轨迹点投影到不同道路上时，通过控制系数ρ，减小状态转移概率值，降低路口处轨迹点投影错误的概率。

（4）基于经验路线的路径规划研究，对北京市热点 OD （Origin and Destination）进行经验路线提取，存储为单独的经验路线图层，采用分层的路径规划算法，将经验路线图层融入到路径规划，形成一种基于经验路线的路径规划方法。

图 1-1 本文研究的主要内容

Figure 1-1 The main content of this paper

1.2.4 论文组织结构

本文主要一共 5 个章节，文章结构如下：

第 1 章，绪论。首先介绍论文研究方向选题的背景、意义，然后分别阐述了浮动车技术研究现状和路径规划研究现状，基于现在研究存在的问题确定本文的主要研究内容，最后对本文的组织结构进行简要说明。

第 2 章，浮动车轨迹数据预处理与特征分析。首先对浮动车轨迹数据进行预处理，然后对轨迹的时空特征分布进行分析，最后对轨迹的行为特征进行分析，识别路网信息和交通状态信息，为下一步研究提供参考。

第 3 章，基于浮动车轨迹数据的经验路线挖掘。结合出租车的轨迹数据和用户上下车数据提取载客状态的车辆运行轨迹，根据轨迹相似度进行聚类，提取质心经验轨迹。使用改进的 HMM 算法，实现轨迹与地图数据的精确匹配，并在地图数据中提取经验路线的道路序列。

第 4 章，基于经验路线的分层路径规划。将计算的经验路线存储为单独的图层，结合 A*算法实现基于经验路线的分层路径规划。

第 5 章，总结与展望。对本文的研究内容和成果进行总结，指出研究的不足之处，对今后需要研究的方向做出展望。

1.3 本章小结

本章对论文的研究背景和意义进行了介绍，详细分析了浮动车技术和路径规划国内外研究现状，基于目前研究存在的问题提出本文主要研究内容，并对论文的组织结构进行了介绍。

2 浮动车轨迹数据预处理与特征分析

据统计滴滴出行平台上的交通工具每天甚至可以将北京所有道路覆盖 300 多遍，出行大数据的轨迹数据体现了时间和空间上的流动性，蕴含了丰富的轨迹特征。对浮动车轨迹数据进行分析，宏观上研究轨迹数据的时空特征可以反映人们的出行规律及出行特征，微观上车辆的行为轨迹实时反映了实地的路网信息和交通状态信息。研究轨迹数据时空特征和行为特征，可以为路径规划提供实际参考信息。

本章主要从原始轨迹数据和用户上下车数据出发，对数据进行清洗去噪、轨迹穿串、提取载客状态的数据。通过对载客量数据进行统计分析，发现出行规律，可以帮助人们合理规划出行时间，避开出行高峰期；对载客时长和载客距离进行分析可以为第四章中经验路线图层提取的路径长度提供参考；利用车流量及速度随时间变化的关系可以进行更加准确的分时路径规划，车流量及速度分析不仅有助于路径规划研究，也广泛应用于智能交通的各个方面，如交通异常探测，短时交通预测，拥堵感知等。另一方面，浮动车轨迹数据蕴涵了车辆的行为特征，车辆的行为轨迹是司机在遵从交通法规的基础上的行为，对轨迹的行为特征进行分析挖掘，匹配不同的行为模式，可以识别实地的交通信息，如交通限制、限速电子眼等。交通限制、电子眼等交通信息是参与路径规划的重要因素，通过轨迹行为特征识别交通信息比传统的电子地图数据具有更高的时效性和准确性。交通信息的识别既可以完善地图数据，又可以参与第四章的路径规划中，提高路径规划的准确性。

2.1 数据预处理

2.1.1 数据准备

本文采用北京市 2018 年 4 月 10 日至 2018 年 4 月 25 日，16 天的滴滴轨迹与用户上下车数据进行分析，包含两个周末和完整两周工作日的时间。

浮动车 GPS 轨迹数据记录了司机用户 ID、经纬度坐标、位置点时间、速度、方向、坐标系统、载客状态等信息。用户上下车数据记录了乘客用户 ID、使用者上车位置信息、使用者下车位置信息、上车时间、下车时间等信息。详细数据结构说明和数据样例如表 2-1 至表 2-4 所示。

表 2-1 GPS 轨迹数据说明

Table 2-1 Field description of GPS data

编号	字段名称	字段类型	字段描述
1	司机用户 ID	string	车载终端 GPS 设备获取，匿名的用户 ID
2	经度	double	保留小数点后 5 位小数，四舍五入
3	纬度	double	保留小数点后 5 位小数，四舍五入
4	位置点时间	long	位置点产生时间，记录方式为时间戳，单位：秒
5	速度	double	保留小数点后一位，四舍五入，单位：米每秒
6	方向	double	记录行驶方向，一般通过度来表达（0-359 度）
7	坐标系统	string	空间位置（坐标）的参照系
8	载客状态	string	车辆的载客状态

表 2-2 原始 GPS 轨迹数据

Table2-2 Original data of GPS

司机用户 ID	经度	纬度	位置点时间	速度	方向	坐标系统	载客状态
56506754	121.5921	29.90653	1460322298	15.4	62.2	GCJ_02	null
563753	120.7225	27.98532	1460322298	-1	-1	GCJ_02	null
8787047	121.6927	31.05259	1460322298	11.8	171.3	GCJ_02	null
4652470	112.5917	37.79378	1460322298	18.4	269.2	GCJ_02	null
4920467	105.1819	37.51381	1460322298	0	29.1	GCJ_02	null
3169414	121.5024	31.27402	1460322298	12.5	133.3	GCJ_02	null

表 2-3 用户上下车数据说明

Table 2-3 Field description of passengers' data

编号	字段名称	字段类型	字段描述
1	乘客用户 ID	string	手机等移动终端 GPS 设备获取，匿名的用户 ID
2	上车经度	double	位置点坐标，保留 5 位小数，采用 GCJ-02 坐标系
3	上车纬度	double	位置点坐标，保留 5 位小数，采用 GCJ-02 坐标系
4	下车经度	double	位置点坐标，保留 5 位小数，采用 GCJ-02 坐标系
5	下车纬度	double	位置点坐标，保留 5 位小数，采用 GCJ-02 坐标系
6	上车时间	string	记录方式为年-月-日 小时：分钟：秒
7	下车时间	string	记录方式为年-月-日 小时：分钟：秒

表 2-4 用户上下车数据

Table 2-4 Original data of passenger

乘客用户 ID	上车经度	上车纬度	下车经度	下车纬度	上车时间	下车时间
20195858	104.0074	30.5486	104.0941	30.6891	2018/4/16 19:47	2018/4/16 20:37
1047689	116.3225	39.8942	116.4586	39.8990	2018/4/16 21:35	2018/4/16 22:12
102336	117.2787	39.1321	117.2326	39.1583	2018/4/16 14:30	2018/4/16 14:51
190794	116.2239	39.9294	116.3213	39.8959	2018/4/16 5:53	2018/4/16 6:05
9249451	120.3484	36.1122	120.3951	36.0598	2018/4/16 11:28	2018/4/16 11:54
501755	120.1709	30.3045	120.1983	30.2320	2018/4/16 12:27	2018/4/16 12:52
16466	116.6599	40.1619	116.6595	40.1806	2018/4/16 22:03	2018/4/16 22:09
206600	116.7162	23.3706	116.6961	23.3745	2018/4/16 2:17	2018/4/16 2:23
127851	120.6864	31.2946	120.7108	31.3416	2018/4/16 12:52	2018/4/16 13:10

2.1.2 数据预处理

（1）数据预处理

因 GPS 精度、系统误差、信号强弱及周围环境等影响，浮动车轨迹数据中存在部分异常数据，异常数据会对后续的数据处理结果产生一定影响，使用前需进行数据清洗。数据清洗内容主要包括：剔除因信号等原因产生的非完整字段和非完整行记录数据；剔除速度、角度过大的异常点，定义速度超过 41.7m/s（150km/h）或角度超多 360 时为噪音点；剔除车辆在停止状态时，速度为 0 的重复数据；剔除因信号不稳定同一车辆同一时间下生成的多条冗余数据。GPS 轨迹数据中对时间的记录采用了 unix 时间戳，unix 时间戳定义为从格林威治时间 1970 年 01 月 01 日 00 时 00 分 00 秒起至现在的总秒数。我们进行时间数据处理时，一般基于标准时间格式，且为了与用户上下车数据格式一致，方便后面的计算，这里将原始 GPS 轨迹数据中的时间数据统一转换为标准时间格式的北京时间。

（2）轨迹穿串

如表 2-1 中所示，原始轨迹数据是按分散的点进行存储，所有车辆的轨迹点混合存储，为方便研究将 GPS 轨迹数据依据用户 ID 进行分组，同一用户的轨迹点数据根据位置点时间进行排序，生成连续的用户完整轨迹串。轨迹串数据的存储结构 $T_i = \{ID : P_1; P_2; \cdots\cdots; P_n\}$ $P_j = \{x, y, timestamp, speed, direction, coordinate\ system\}$。当相邻轨迹点时间差大于 10 分钟时，切断轨迹，生成该用户的一条新轨迹串。

（3）载客状态下的轨迹数据提取

浮动车轨迹中非载客状态下的轨迹目的性较弱，无研究意义，所以应该提取载客状态下的的轨迹数据作为研究对象。由于原轨迹数据中载客状态字段不可用，结合用户上下车数据进行匹配，提取载客状态的轨迹。对用户上下车数据中的上下车位置和时间分别设置阈值，在穿串后的轨迹串数据中，匹配阈值范围内的同一司机用户的起终点轨迹点数据，并提取起终点轨迹间的连续轨迹子串数据，作为一条完整的载客轨迹。

2.2 浮动车轨迹数据时空特征分析

浮动车轨迹数据从不同的维度、不同的粒度反映着人们的出行规律和出行特征。浮动车轨迹是地方交通的真实轨迹，同时也反映了所在城市的地方交通特征。通过对轨迹数据的时空特征分析，了解人们出行特征和地方交通特征，为规划出行方式和路线选择提供参考依据。

轨迹数据具有时空性，时空数据反映了空间数据随时间变化的趋势，具有时序性。由于人们作息在时间上呈现规律性，所以出行轨迹数据在时间上亦反映出周周期与日周期的特征。

2.2.1 载客数据分析

（1）载客量分析

统计北京一定区域（10km*20km）16 天的载客数据,对周期趋势进行分析。

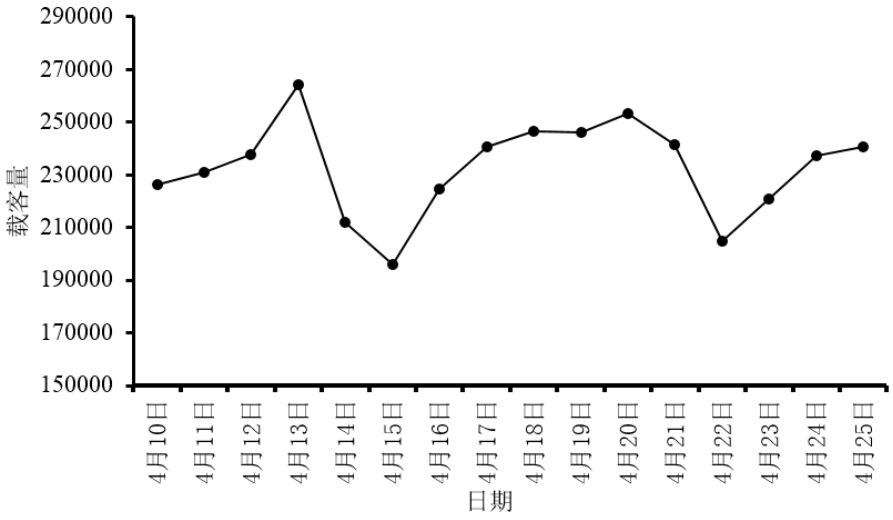


图 2-1 载客量周周期趋势分析

Figure 2-1 Weekly-cycle property of the number of taxi trips

图 2-1 载客量数据呈现以周为周期的趋势，4 月 14 日、4 月 15 日、4 月 21 日、4 月 22 日为周末，周末载客量较工作日有减少，其中周日载客量比周六有减少。工作日中周五载客量最多，周五为工作日的最后一天，一周的工作和学习结束，很多人会抓住周五这个时间点，选择走出来放松，聚会娱乐。

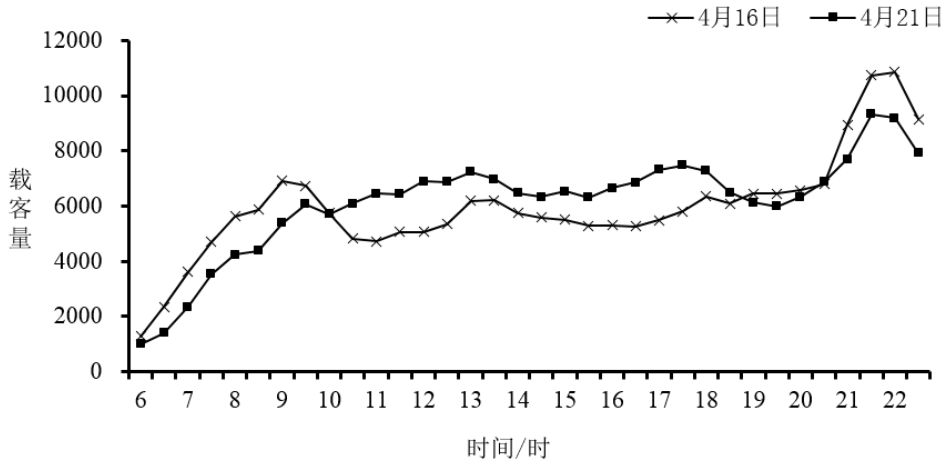


图 2-2 载客量日周期趋势分析

Figure 2-2 Daily-cycle property of the number of taxi trips

图 2-2 载客量的日周期趋势中工作日早九点左右会出现高峰，周末没有早高峰，出行时间分散。晚上 9 点半到 10 点半工作日和周末载客量都相对较大，主要因素是工作日晚上加班打车和周末出行晚上打车回家。总体来看，工作日的周期性特征比周末更为突出些。

（2）载客时长分析

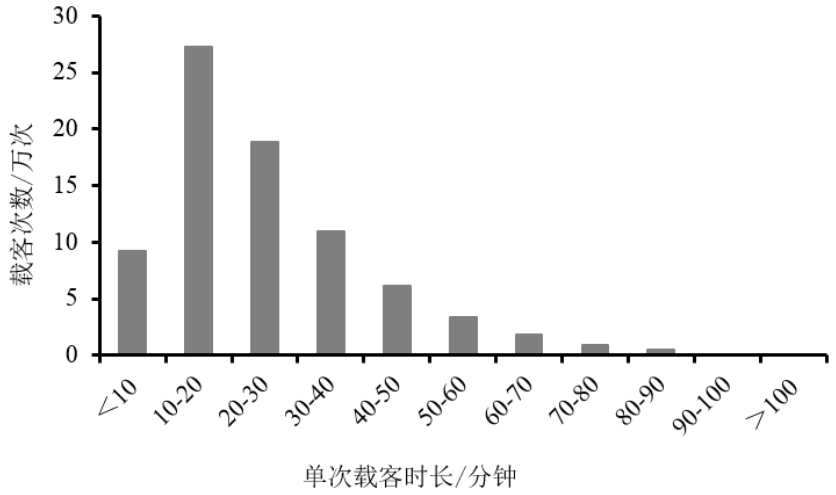


图 2-3 每日载客时长分布

Figure 2-3 Daily travel time distribution

图 2-3 中，单次的载客时长最多的是 10-20 分钟，其次为 20-30 分钟，占总载

客次数的一半以上，随载客时长的增加，载客数量逐渐减少，载客时长 1 小时以内的次数占总载客次数的 95%以上。

（3）载客距离分析

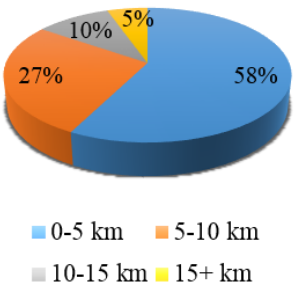


图 2-4 载客距离分析

Figure 2-4 Percentage of travel distance

载客距离反映了居民出行距离，为城市交通规划布局提供重要参考。载客距离是通过乘客上下车位置的经纬度计算起点和目的地间的实地直线距离，这里使用测地距离来计算地球椭球面模型表面上最短的距离。

将载客距离以每 5km 进行区间划分，分别统计一周内各载客区间段的载客量占总载客量的比例。如上图 2-4 所示，载客距离 5km 内的载客量占比 58%，超过总载客量的一半，随载客距离的增长 5-10km、10-15km、15km 以上的比例呈现递减趋势，符合居民出行特点。

2.2.2 车流量及速度分析

（1）车流量分析

通过统计北京三元桥附近从早上 6：00 到晚上 22：30 间，每半小时内的轨迹车流量，如图 2-5 所示可以看出早上 9 点半左右迎来早高峰，晚上 7 点迎来晚高峰。工作日客流量比周末要大很多，早高峰在周末时不突出，与周末上班族不出勤有关系。图 2-5、图 2-6 中分别选取职、住区域的车流量进行分析，发现住宅区早高峰比商业办公区要早，存在迁徙时间差，据 2018 年中国城市通勤研究报告统计，2018 年北京市平均通勤时间为 56 分钟。

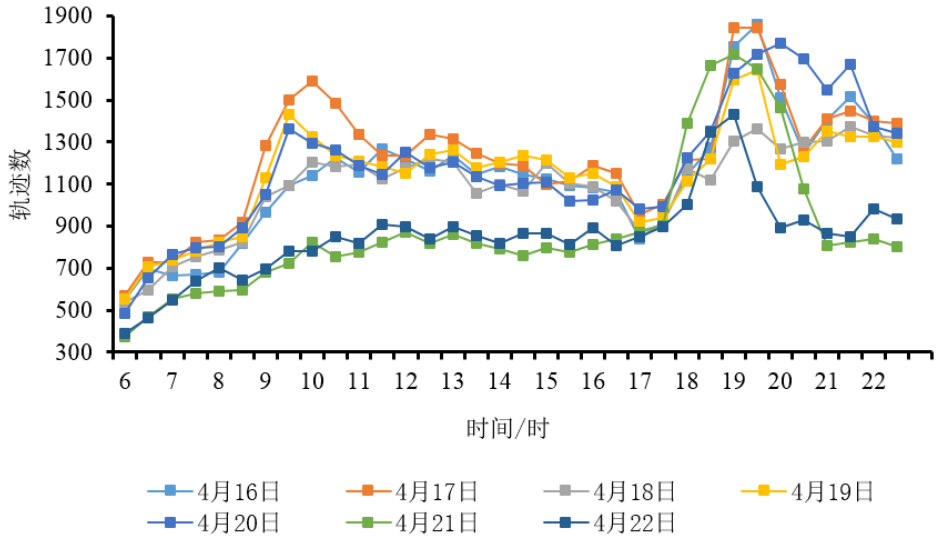


图 2-5 每日不同时刻车流量变化（三元桥附近）

Figure 2-5 Changes of traffic flow near San Yuan Qiao

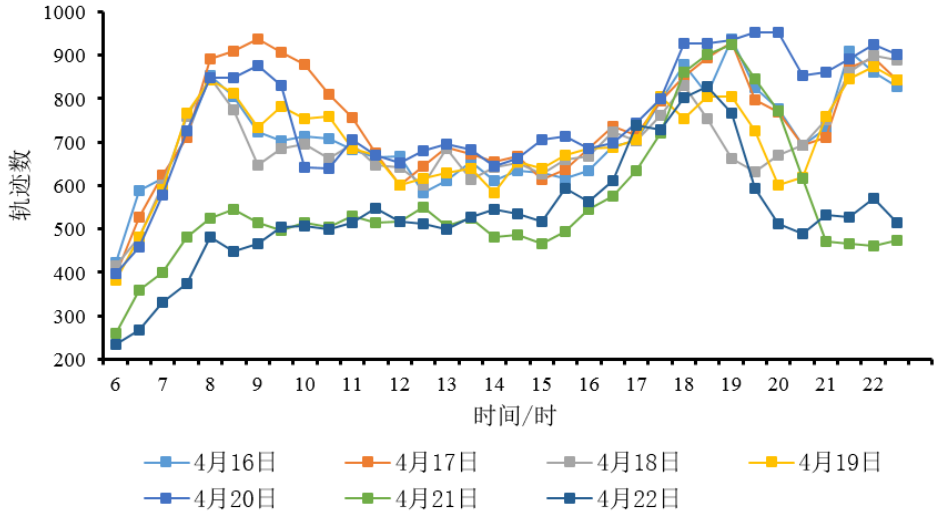


图 2-6 每日不同时刻车流量变化（榴乡桥住宅区附近）

Figure 2-6 Changes of traffic flow near Liu Xiang Qiao

（2）速度分析

选取某一地点，计算一天中早上 6：00 到晚上 22：30 间，每半小时的平均速度，如图 2-7。可以看出速度的变化与车流量基本相反，同样存在早、晚高峰的特征，早 7 点-10 点行驶速度较低，晚上 6 点左右晚高峰时行驶速度明显下降，与出行规律相符。研究历史轨迹的平均行驶速度和速度的变化，可以为路径规划的路径选择提供依据，并估算预计到达时间。

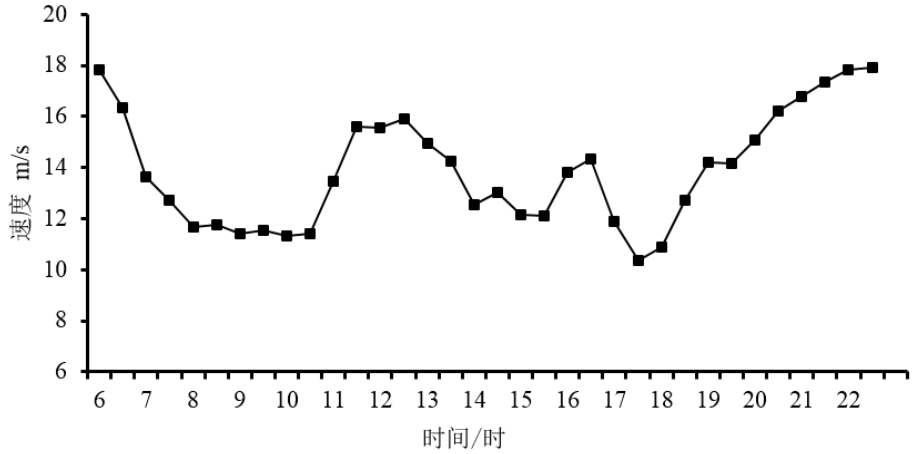


图 2-7 同一地点一天中不同时刻车流平均速度变化

Figure 2-7 Changes of average traffic speed in the same location varies of the day

2.3 浮动车轨迹数据行为特征分析

通过对海量轨迹数据的行为特征进行挖掘分析，提取有效的交通信息，可以实时发现地图数据的变化，包括道路施工解除、道路交通限制变更、道路方向变化、道路最高速度限制变化、卡车经验路线、联网安全服务、兴趣点变化及热度等信息。现实世界中，城市地表地物飞速变化，通过对轨迹数据挖掘可以实现导航电子地图的快速更新，解决传统导航地图更新慢、现势性低、覆盖度不够的问题。另一方面，道路限制信息、路口平均等待时间、道路平均行驶速度等，这些实时更新的交通信息，参与到路径规划算法中，能提高路径规划的准确性。

近年来，基于浮动车轨迹数据的交通信息挖掘逐渐被各位学者关注,学者们提出不同方法对各种交通信息进行挖掘。对地图进行网格化，统计网格内道路段的方向信息，过滤掉存在两个大量的相反方向的路段，可以发现疑似单向限制道路[30]。对转弯曲线轨迹数据进行聚类，轨迹密集区可以识别路口区域范围[31]。利用混合高斯模型对轨迹数据进行建模，提取车道数量等[32]。交通信息的识别依赖车辆的运动行为模式，轨迹的诸多特征包括经纬度位置信息、速度、方向及时间等可以很好的描述车辆的运动行为。所以本文对轨迹进行特征选择，通过模式匹配的方法对车辆的行为模式进行匹配，进而识别出各种交通信息。

2.3.1 轨迹行为特征分析技术流程

时空轨迹序列记录了对象连续的移动行为，浮动车的运动行为模式可以反映实地的交通信息，通过选择轨迹特征，训练分类器，将轨迹按行为特征进行分类，

同时定义可以表述不同交通信息的各种行为模式，将分类好的轨迹行为进行交通模式匹配，进而识别当地的交通信息。轨迹行为特征分析的主要流程如图 2-8 所示：

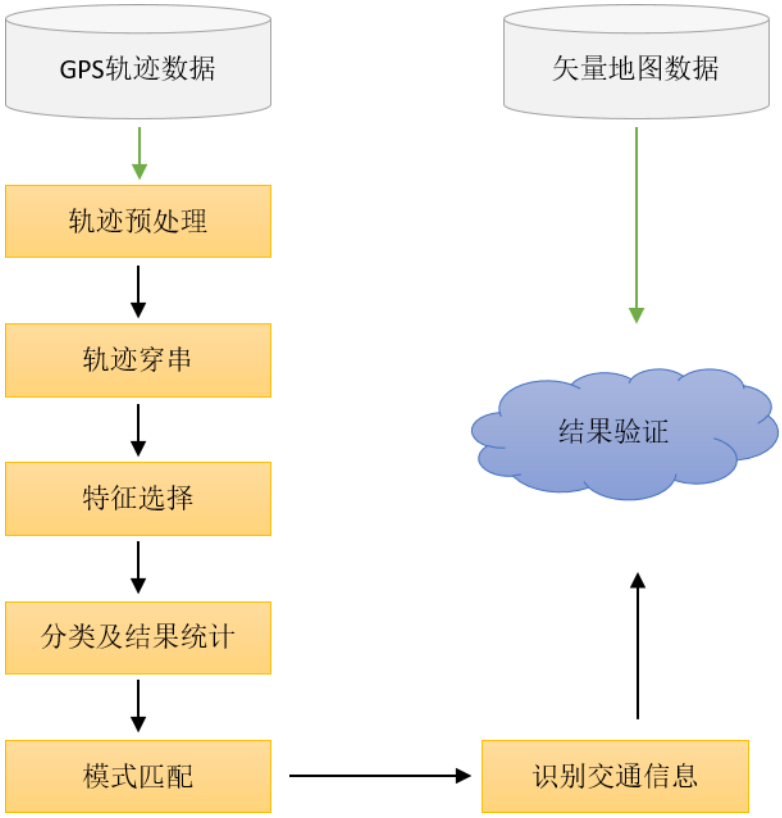


图 2-8 交通信息识别技术流程

Figure 2-8 Technical flowchar of traffic information identification

2.3.2 基于轨迹行为特征的电子眼限速值识别

电子眼限制的速度值是路径规划进行道路选择的重要前提，也是导航电子地图的重要数据要素。目前导航电子地图数据中使用的摄像头数据均是由人工采集验证。这种方式在城市发展初期比较有效，随着城市规模的扩大，城市道路里程快速增长，摄像头数据量呈现出几何式增长。传统的地毯式采集、更新方法已经跟不上数据更新速度，不能有效为用户提供更加准确的服务，甚至由于摄像头数据的陈旧，会对城市道路的畅通起到相反作用。比如电子地图中一条道路限速均为 120km/h,如果电子地图错误的限速为 100km/h，必将会导致司机的突然减速，而这种错误的数据会误导司机，甚至会造成交通拥堵以及引发交通事故。再者，一些新增的摄像头数据由于未及时更新到电子地图中，会造成地图使用者的交通违章，给用户造成经济损失。针对基于轨迹的行为特征识别电子眼限速值这一技术进行探索和实践，通过分析轨迹点数据的特征，得到开车人员的增减速的行为。

再采用聚类、匹配等数据分析手段，分析开车人员的行为，结合车速信息，快速提取电子眼变化，识别限速信息，更新电子眼数据。

以轨迹数据为基础，分析出基于瞬时点的司机驾驶行为。在司机行为数据的基础上进行历史摄像头数据快速更新以及新增摄像头数据挖掘。

（1）识别新增摄像头数据

在轨迹点数据集中去除历史摄像头位置前后缓冲区的数据，在剩余的轨迹点数据中，对驾驶行为数据进行连续差分。如果有符合电子眼位置的行为特征的轨迹数据且到达相应比例，则将位置标记为可能新增电子眼地点，反向验证其他用户轨迹数据，若能达到相应比例，则标识为特征变化点，需要进一步对所得数据进行实地验证，从而决定是否新增为电子眼数据。

司机的行为特征判定。首先，对司机进行类别划分，计算路段平均速度和司机平均行驶速度，将大于路段平均速度 20%以上的司机划分为快速型 S 型，低于平均速度 20%以上的司机划分为低速型，中间部分划分为中速型 M 型。其次，识别加减速行为，对司机行驶速度进行连续差分，计算加速度 a，对加速度有突然变化点进行标记记录加减速行为。然后，对突然减后速突然加速行为模式进行识别，分别对快速型和中速型司机设定减速距离 L_1 和减速时间 t_1，加速距离 L_2 和加速时间 t_2，快速型和中速型司机的行为判别模式分别为 S{*S_L1，S_t1，S_a1，S_L2，S_t2，S_a2*}与 M{*M_L1，M_t1，M_a1，M_L2，M_t2，M_a2*}。

根据上述的行为特征模式对数据进行聚类，提取可能新增电子眼位置如图 2-9 所示，判断该位置限速值，用原有地图数据中的电子眼数据对结果进行验证。

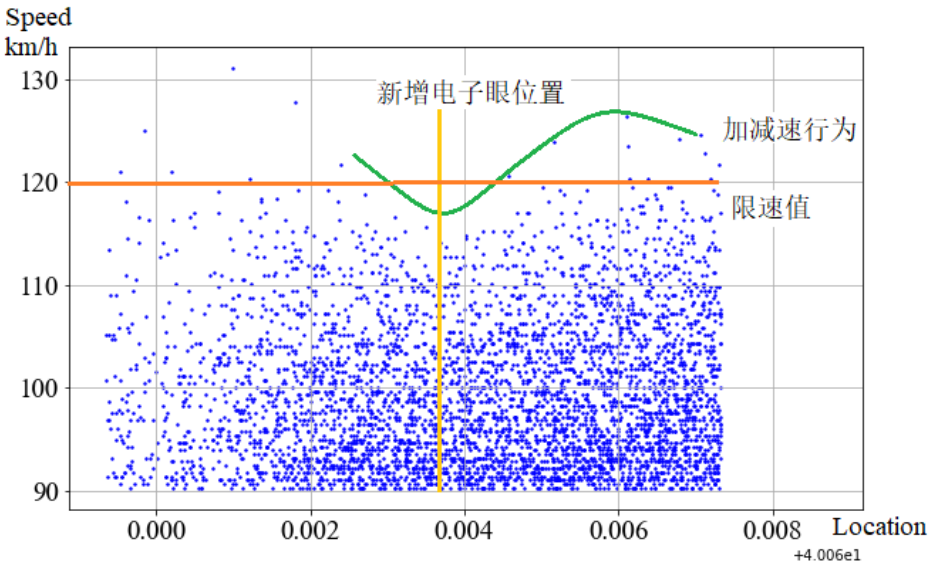


图 2-9 电子眼位置示意图

Figure 2-9 Position electronic eye

（2）历史电子眼数据更新

历史电子眼前后缓冲区内数据经过决策树分类和频度统计，得到不同用户在同一点的速度分布情况和历史电子眼限速值。根据轨迹点的速度概率，设定速度阈值。判断估算速度值和历史数据值的关系，如果二者相差在设定阈值范围内，则认为无需更新，如果超过设定阈值，则将该限速点设为需要实地验证点。进一步进行实地验证，最终根据实际情况是否更新限速信息。

2.3.3 基于轨迹行为特征的交通限制识别

交通限制是为保持车辆在道路上安全行驶所规定的限制，是进行路径规划的重要约束条件，包括道路上车辆的行驶速度限制和转向限制。道路限速目的是保障车辆在不同等级道路、不同路况的道路上的安全行驶速度。转向限制一般为合理规划城市交通、缓解特定路口的交通压力、疏解拥堵、维持交通秩序，限制车辆左转或者掉头等，包括禁止掉头、禁止左转、禁止直行、禁止右转，限制信息包括永久限制和时间段限制。很多时间段交通限制是交管部门根据实际交通状态增加的限制，转向限制一般以路边标牌的形式进行提示，但新增加的限制标牌人们往往容易忽视，且地图数据更新需要时间，司机不能及时得到有效的交通提示。转向交限 80%会设置的路口处，路口一般会设有电子违章拍照，这种情况下往往也会给司机带来经济损失。据了解 2018 年北京市北五环东西向广顺桥出口在广顺桥路口处设置了时间段禁左交限，广顺桥路口南向北也设置了时间段禁左交限，很多司机在此处因违反禁令标志指示受到违章处罚。当增加实地交通限制标牌时，部分司机已经根据禁令调整自己的行车路线，通过直行掉头或者右转掉头等方式绕行至目的地，我们可以通过对浮动车轨迹行为模式进行统计分析，识别这种间接的行为模式，进而得到交通信息的变化。

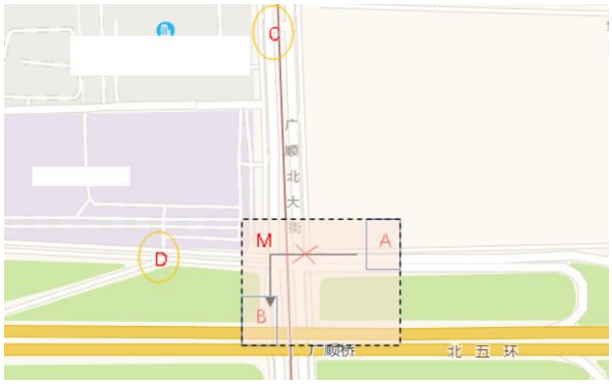


图 2-10 广顺桥禁左交限示意图

Figure 2-10 No left turn of Guang Shuan Qiao

以图 2-10 广顺桥路口西向南方向禁左交限为例，根据此处的轨迹行为特征判断是否存在禁左交限，步骤如下。

（1）选择轨迹驶入、驶出路口的区域范围 A、B，确定驶入、驶出方向（需要识别的交通限制方向），提取符合驶入、驶出区域范围和方向的完整轨迹。

（2）方向数据离散化，将行驶轨迹点方向的角度值按表 2-6 对应到 8 个方向值。

表 2-6 方向数据离散化

Table 2-6 Discretization of direction data

角度值	方向	方向说明
338-359,0-22	1	北向行驶
23-67	2	东北方向行驶
68-112	3	东向行驶
113-157	4	东南方向行驶
158-202	5	南向行驶
203-247	6	西南方向行驶
270-292	7	西向行驶
293-337	8	西北方向行驶

（3）轨迹行为特征提取，对上述步骤提取的每一条经过路口的轨迹，提取主要特征，包括轨迹驶入方向、轨迹驶出方向、轨迹方向值分布、轨迹位移差变化趋势、轨迹点坐标分布。轨迹点方向值分布，需统计每条轨迹的轨迹点在 8 个方向值中的比例，关注是否存在较大比例的两个相反方向值存在。轨迹点位移为距离起终点的位移差，关注是否存在同时逐渐增大，又同时逐渐减少。轨迹点坐标需关注与直接左转的坐标覆盖范围 M 的关系，确定直接左转的轨迹坐标覆盖范围 M（图中虚线所示），对于此处西向南左转轨迹范围主要确定纵坐标的最大值和横坐标的最小值，即 A 区域的 max(y)，和 B 区域的 min(x)，统计每条轨迹中横坐标小于 min(x)的比例，和纵坐标大于 max(y)的比例。

（4）用决策树算法对特征进行分类，最终将轨迹行为分为 3 类，特征分类过程如图 2-11 所示。直接左转情况：A-B；右转掉头的情况：A-C-B；直行掉头后右转的情况：A-D-B。后两种情况绕远左转轨迹如图 2-12 所示。

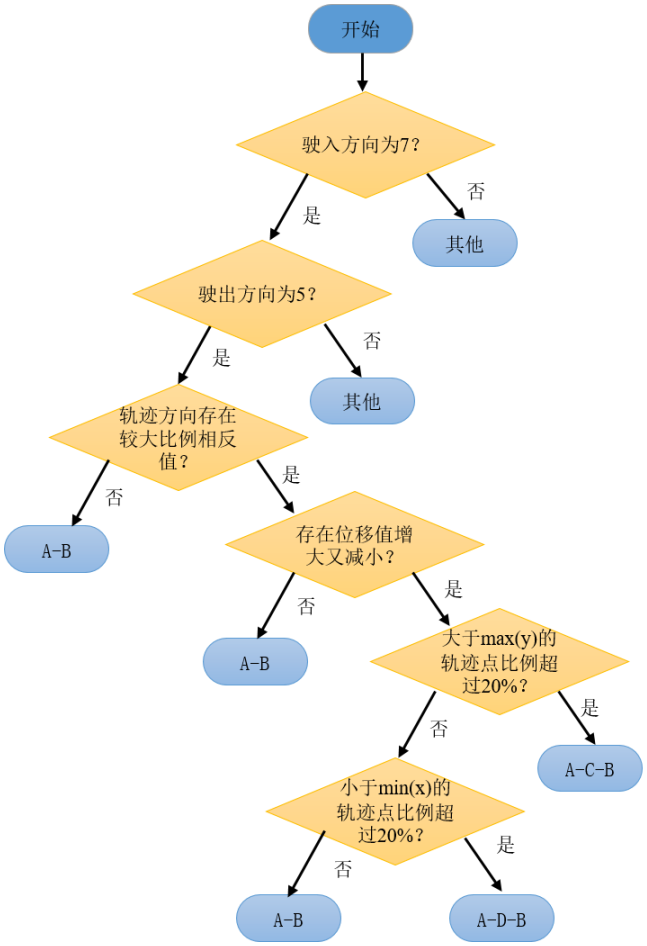


图 2-11 决策树特征分类

Figure 2-11 Feature classification by Decision tree

（5）分别统计三类行为轨迹的占比，A-B：20%；A-C-B：50%；A-D-B：30%。设置阈值，当间接左转 A-C-B 和 A-D-B 占比总和>50%时，判断现场可能存在交通限制，标记新增禁止左转交通信息，设为需要实地验证点。进一步进行实地验证，最终根据实际情况是否更新交通信息。

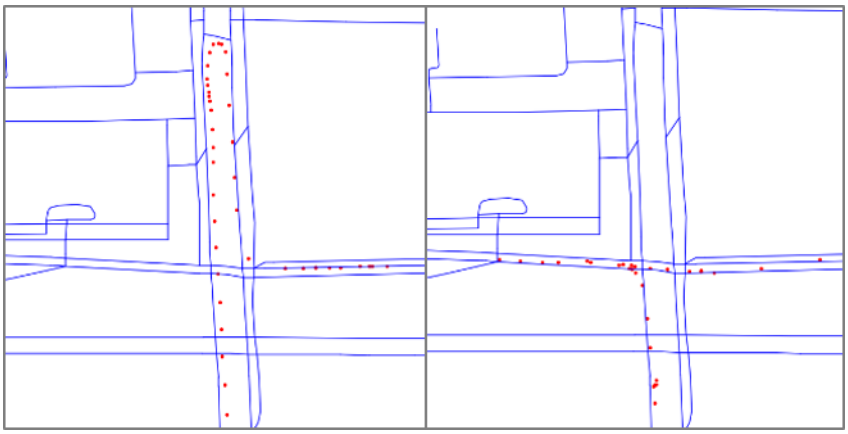


图 2-12 绕远左转轨迹示意图

Figure 2-12 Roundabout route of turn left GPS

2.4 本章小结

本章对论文使用的数据进行了详细说明，包括数据范围、数据时间及轨迹数据和上下车数据的数据格式和详细字段描述。数据使用之前，对数据进行了清洗去噪、轨迹穿串、载客状态下轨迹的提取等处理，将数据整理成标准的结构进行存储，方便后续使用。

其次，通过对轨迹数据进行统计分析，得到交通轨迹的时空分布特征，了解人们出行特征和地方交通特征，对载客量、车流量分析发现周周期、日周期的周期特征；职住区域早晚高峰存在时间差；载客距离 10km 以内占 85%；载客时长主要分布在 10-30 分钟等信息为交通信息挖掘、规划出行方式和路线选择提供参考依据。

本章在研究轨迹时空特征的基础上，对轨迹的行为特征进行分析，提出基于特征提取的模式匹配方法进行交通信息识别。分别就电子眼限速值和禁止左转两种交通信息的挖掘做了详细的方法描述。基于轨迹距离、轨迹时间、轨迹加速度的特征对司机加、减速行为模式进行识别，进而判别限速电子眼的存在和具体限速值。基于轨迹进入、退出路口的位置、方向，轨迹点方向分布，轨迹点坐标分布，轨迹位移差变化趋势，识别司机在路口的绕远和掉头行为，进而判别是否存在交通限制信息。通过对轨迹行为特征进行分析，能有效识别交通信息，对导航电子地图数据的更新和路径规划有重大意义。

3 基于浮动车轨迹数据的经验路线挖掘

浮动车数据采集的主要来源是出租车，出租车是城市人群运移的重要工具，因此出租车司机往往是对道路结构、路况信息最为熟悉的群体。载客状态下的浮动车轨迹更具有很强的目的性，司机为了能尽快将乘客送达目的地，他们根据出行经验，选择最快最优的路径，所以出租车司机的行驶路线也被人们称为“经验路线”。经验路线蕴含着司机的驾驶经验智慧，综合考虑了道路等级、红绿灯数量、拥堵程度及周围环境等各种因素，比传统的路径规划算法更有优势。通过对浮动车轨迹进行挖掘，提取出最大程度上包含司机典型经验的完整路线，并将经验路线融入到道路路径规划中，建立合适的路径规划模型，对车辆出行具有重要的指导意义。

本章主要研究经验路线的挖掘方法。首先，对起终点 OD 间的所有轨迹点进行 DBSCAN 聚类，得到最大类轨迹点，即选择最多的路线轨迹，对最大类轨迹点根据轨迹间相似度进行 k-medoids 聚类，过滤掉选择偏少的分支轨迹和有绕路的噪声轨迹，提取 OD 间质心轨迹。然后，采用一种基于改进隐马可夫模型的算法，对质心轨迹与道路地图数据进行完整匹配，得到地图上完整的经验路线 LINK 序列。

3.1 经验路线挖掘流程

在大规模的轨迹数据中提取司机行驶较为频繁的经验路线的整体流程如图 3-1 所示，步骤“轨迹聚类生成最大轨迹簇”、“轨迹聚类提取质心轨迹”的具体方法在 3.2 节中进行描述，步骤“地图匹配”的具体实现方法在 3.3 节进行描述，步骤“生成经验路线”的具体方法在 3.4 节中进行描述。

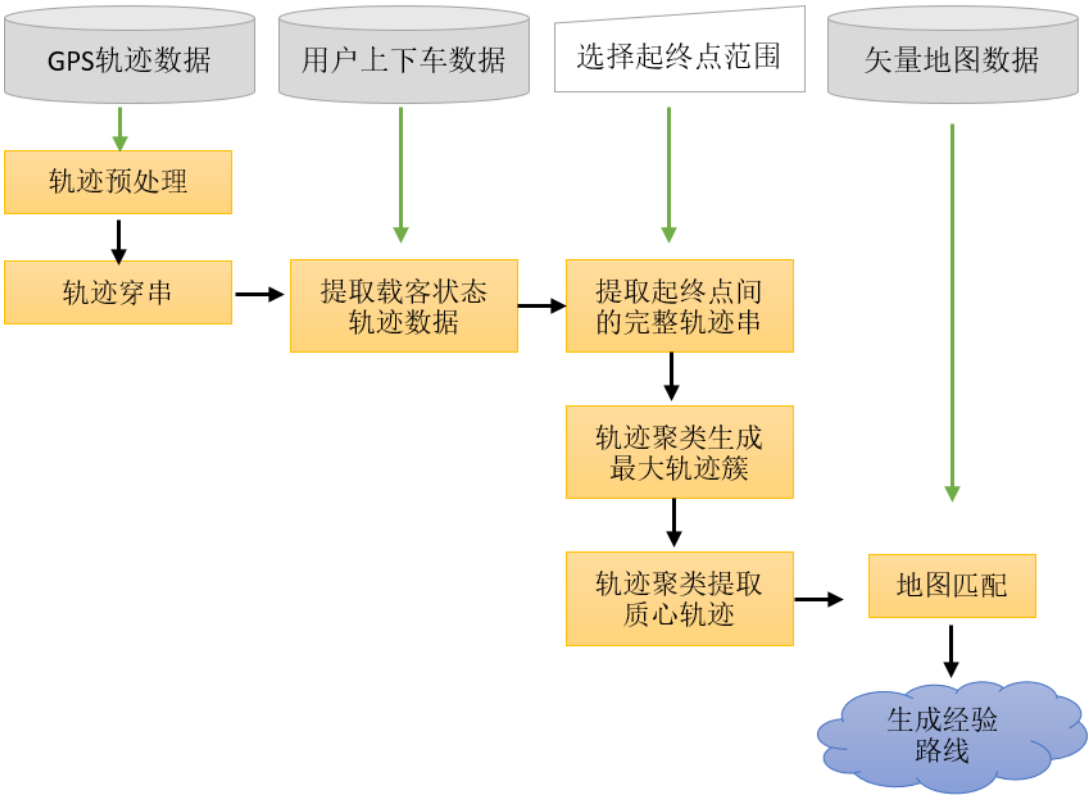


图 3-1 经验路线提取流程

Figure 3-1 Technical flow of extract path with experience

3.2 轨迹聚类

3.2.1 常见聚类方法

空间聚类分析是将空间数据集按一定的规则，划分成若干个有意义或有用的类簇，并使同一类簇内数据对象的相似度最大，而不同类簇间的数据对象相似度最小，其主要方法有划分聚类、层次聚类、网格方法、密度方法及模型方法 5 大类[34]。

划分聚类，输入内容为一组包含 n 个不同元素的数据集，聚类前需要人为指定这些数据集需要划分的分组数 k，这些分组代表聚类的结果，其中 k 的值必须小于数据集的元素个数 n。例如：k-means 聚类，比较简单且概念容易理解，由于实现过程简洁明了、计算速度快，使得它成为应用最为广泛的聚类之一。k-means 聚类开始随机从输入数据集中选取 k 个聚类中心点，接着遍历数据集中的每一个元素，计算这些元素与初始聚类中心点的距离，选取其中最近的距离对应的聚类中心点，将该元素加入到对应的簇，遍历完成后对 k 个簇重新计算中心点，循环这个过程直到中心点没有变化。这类聚类方法的缺点是只适合特定分布的数据。

层次聚类，这种方法又称为系统聚类。比较典型的用法是对植物进行分类。聚类过程从具体的植物物种开始，对所有植物按照相似度进行分层似的分类，以各种植物王国结束，每个植物王国里面包含了很多更小的簇。层次法聚类的结果不是具体的分类，而是相应的树状图分簇，输入数据的信息以树的结构展示出来。这样的特点导致层次聚类法的适用范围有很大限制，没有层次的数据集无法使用。层次聚类的计算过程很复杂，循环次数非常多，导致性能差效率低，整个聚类处理的流程耗时很多。还有另一个重要的缺陷，最终聚类的树状结构的结果并不精确，导致这个聚类算法使用场景比较有限。

基于网格的方法，首先将输入数据所在的空间按照一定的规则划分成为网格结构，遍历所有网格，对每个网格内的数据进行统计，根据这些统计信息，把相连的网格聚类成一个簇。基于网格的算法优点很明显：聚类时处理的是单个网格不依赖输入数据的个数，因此时间复杂度低于基于密度的算法，速度快；另外一个就是能处理非凸边形的数据。相应的缺点也很明显：对事先设置的参数敏感非常高，分布不规则的数据也不太适用；处理单元是基于网格而不是基于每个输入数据，导致精确性也相对较差。有些情况可以结合基于密度的方法一起使用，以克服上述缺点。

基于模型的方法，主要有两种，一种是基于概率模型的方法、另一种是基于神经网络模型的方法。基于模型的方法给每一个聚类假定一个模型，然后去寻找能够很好的满足这个模型的数据集。优点是聚类结果不局限于类圆形，也支持椭圆形；对簇的划分也没有那么强制性，一个数据点也可以同时属于多个簇，例如聚类结果中可以出现数据点 N 属于第 1 个簇的概率为 30%，属于另一个簇的概率为 70%。

基于密度的方法，主要基于数据集中数据的密度不同对数据进行分类，考察样本之间的可连接性，并基于可连接样本不断扩展聚类簇，用低密度区域分割高密度区域实现数据的聚类分析，算法主要有 DBSCAN[33]与 OPTICS[34]算法等，能发现任意形状的聚簇，聚类结果几乎不依赖于结点遍历顺序，能够有效的发现噪声点。文献[35]通过对浮动车轨迹数据进行密度聚类，提取驾驶员的驾驶兴趣区域信息；文献[36]提出了一种轨迹数据聚类的 NETSCAN 算法，可得到在交通路网约束条件下路网之间的连通关系。文献[37]对移动目标对象的时空数据进行密度聚类，研究目标的移动行为特征。本文使用 DBSCAN 聚类算法，与其它方法的不同的地方在于它聚类时不是基于我们设置的各种方法计算获得的距离的，而是基于数据元素之间的密度的。这样的聚类算法适合各种数据分布形状，能解决基于距离的算法仅适用于类圆形分布数据聚类的缺陷。基于密度的算法，不需要我们提前知道或者人工设置簇的数量，缺点是聚类开始前需要人工输入距离 ε 和 *MinPts*

这两个参数，并且这两个参数的设置对最终聚类的结果影响较大，需要多次调试观察。

3.2.2 轨迹间距离的度量

研究轨迹聚类需计算轨迹相似性，轨迹相似性通常用一个距离函数来计算，现行比较常用的轨迹相似性度量指标有多种，而且分别有各自的优势，如何选择不同的轨迹相似性度量是进行轨迹聚类的关键。

轨迹距离首先可以转换为点与轨迹之间的距离，点 P 到轨迹 L 的距离用 P 到 L 上最近轨迹点的距离来表示。

$$D(p,T)=\min_{q\in T} D(p,q) \tag{3-1}$$

轨迹与轨迹之间较为经典的相似度度量指标主要有以下几种。

（1）CPD (Closest-Pair Distance) 是用两条轨迹（T_1,T_2）上点间距离的最小值来表示轨迹间距离。

$$CPD(T_1,T_2)=\min_{p\in T_1,p'\in T_2} D(p,p') \tag{3-2}$$

CPD 距离定义简单，但是容易受到局部极端情况的影响，考虑两条轨迹在某点相交，然而整体情况差异很大，这种情况用 CPD 距离显然不合适。

（2）SPD (Sum-of-Pairs Distance)，使用两条轨迹上对应点间距离总和来表示距离。SPD 距离要求轨迹 A 和轨迹 B 具有相同的轨迹点个数。

$$SPD(T_1,T_2)=\sum_{i=1}^{n} d(a_i,b_i) \tag{3-3}$$

（3）DTW（Dynamic Time Warping）动态时间扭曲法则，改善了 SPD 对两条轨迹的轨迹点个数相同的限制。对轨迹进行局部的拉伸或者缩放，从而可对不同采样频率和不同长度的轨迹进行比较，DWT 距离就是所有最优匹配轨迹点间距离的累加和。使用动态规划的思想，实现了对某些点的重复使用，确保重复使用的点达成的路径最优的，从而较为高效地解决了数据不对齐的问题。

$$DTW(T_1,T_2)=\begin{cases}0, if\ \ n=0\ and\ \ m=0\\ \infty, if\ \ n=0\ or\ \ m=0\\ d(Head(T_1),Head(T_2))+\min\begin{cases}DTW(T_1,Rest(T_2))\\ DTW(Rest(T_1),T_2)\\ DTW(Rest(T_1),Rest(T_2))\end{cases}\end{cases} \tag{3-4}$$

n,m 分别为轨迹 T_1， T_2 的轨迹长度，即轨迹点个数。

（4）LCSS（Longest Common Subsequence）最长公共子序列，主要考虑轨迹之间相似的部分作为轨迹相似性的度量，因此对于一些因匹配距离超过阈值的轨迹点可以跳过，这样的特征使它对噪声具有鲁棒性。但是它不考虑相似子序列之间不相似部分，会导致判断不准确[38]。

$$LCSS(T1,T2)=\begin{cases}0, if\ \ n=0\ or\ \ m=0\\ 1+LCSS(Rest(T_1),Rest(T_2)),\ if\ \ d(Head(T_1),Head(T_2))\\ \le\varepsilon, and\ |n-m|<\delta\\ \max(LCSS(Rest(T_1),T_2),LCSS(T_1,Rest(T_2))), otherwise\end{cases} \tag{3-5}$$

n,m 分别为轨迹 T_1，T_2 的轨迹长度，即轨迹点个数，δ 为整数，ε 为距离阈值。

（5）EDR （Edit Distance on Real Sequence）实序列编辑距离，通过对字符串进行增、删、改等操作，使其中一个字符串与另一个字符串完全相同所需要的最小操作次数，抑制噪音能力较强，度量能力较差[39]。

$$EDR(T_1,T_2)=\begin{cases}n,\ if\ \ m=0\\ m,\ if\ \ n=0\\ \min\begin{cases}EDR(Rest(R),Rest(S))+subcost,\\ EDR(Rest(R),S)+1, EDR(R,Rest(S))+1\end{cases}, otherwise\end{cases} \tag{3-6}$$

$$subcost=\begin{cases}0,\ if\ \ d(Head(T_1),Head(T_2))\le\varepsilon\\ 1,\ otherwise\end{cases} \tag{3-7}$$

n,m 分别为轨迹 T_1， T_2 的轨迹长度，即轨迹点个数。

（6）ERP（Edit Distance with Real Penalty）实补偿编辑距离，结合了 EDR 和 DWT 的优点，通过使用参考点来计算距离。

对应轨迹点间的距离并不是轨迹点到整条轨迹的最短距离，点到线的最短距离应为点到线上投影点的距离。本文采用的距离度量方法，分解为轨迹点到轨迹

线的投影距离，通过计算轨迹所有轨迹点到另一条轨迹线的投影距离的平均距离来衡量轨迹间的相似度。

轨迹$T_a=\{a_1,a_2,\cdots\cdots,a_n\}$，$T_b=\{b_1,b_2,\cdots\cdots,b_m\}$，$a_i$为$T_a$上的轨迹点，$b_j$在$T_b$上的轨迹点，$a_i$在$T_b$上的投影点为$a_i'$，$b_j$在$T_a$上的投影点为$b_j'$，轨迹点到投影点间的距离采用欧式距离进行计算$D(a_i,a_i')$，轨迹间的距离公式表示为：

$$D_{TT}(T_a,T_b)=(\sum_{i=1}^{n}D(a_i,a_i')/n+\sum_{j=1}^{m}D(b_j,b_j')/m)/2 \tag{3-8}$$

这种轨迹间距离度量的优点是不用考虑轨迹点数量，两条轨迹的轨迹点数不需要相同，且不需要存在对应顺序和关系，使用投影距离更加准确。

3.2.3 DBSCAN 聚类生成最大轨迹簇

DBSCAN 算法“邻域”来描述样本分布的紧密程度，根据其邻域内的对象数目定义核心点分布及其密度相连等数据相关性，通过密度可达关系实现空间对象的聚类处理。本实验中，对轨迹数据采用 DBSCAN 进行聚类，去除噪声点并发现最大聚类簇。轨迹数据集定义$T=\{P_1,P_2,\ldots\ldots,P_n\}$，$P_i$ 为轨迹点，定义一组“邻域”参数$(\varepsilon,MinPts)$来刻画轨迹点分布的紧密程度，其中ε为邻域半径，$MinPts$为邻域半径为ε范围内的最少轨迹点个数。定义以下几个概念：

ε-邻域：对于轨迹点P_i 邻域的定义为

$$N_\varepsilon(P_i)=\{P_j\in T\mid dist(P_i,P_j)\le\varepsilon\} \tag{3-9}$$

核心对象：对于任意轨迹点P_i的ε -邻域至少包含$MinPts$个轨迹点，则P_i为一个核心对象，即

$$|N_\varepsilon(P_i)|\ \ge\ MinPts \tag{3-10}$$

密度直达：若P_j位于P_i的ε -邻域内，且P_i是核心对象，则P_j由P_i密度直达。即

$$P_j\in N_\varepsilon(P_i) \tag{3-11}$$

$$|N_\varepsilon(P_i)|\ \ge\ MinPts \tag{3-12}$$

密度可达：对于P_i和P_j，若存在样本序列$x_1,x_2,\cdots\cdots,x_n$，其中$x_1=P_i$，$x_n=P_j$且x_{i+1} 由x_i 密度直达，则P_j由P_i密度可达。

密度相连：对于P_i和P_j，若存在P_k使得P_i和P_j均由P_k密度可达，则P_i和P_j密

度相连。

其算法的主要思想是将所有的轨迹点标记为未聚类的，读取轨迹点数据，通过 ε 和 *MinPts* 找出所有核心轨迹点。以任一核心轨迹点为出发点，找出由密度可达样本生成的聚类轨迹簇，直到所有核心轨迹点都被访问过，簇不再增长为止。

对所有符合起始点和目的地的所有轨迹进行 DBSCAN 聚类，可以发现所有可能路径，通过调整 ε 半径参数、*MinPts* 邻域密度参数，使得最大聚类簇轨迹点数占总轨迹点数量的 40%以上，此时最大轨迹簇轨迹点数据集 D 中可能包含非完整轨迹或绕远轨迹，再对每一条轨迹 T 进行统计，T_D 表示轨迹 T 的轨迹点在最大簇 D 中的数量，T_D/T 小于 60%时，将轨迹 T 的轨迹点从最大轨迹簇中去除，这时的最大轨迹簇包含的是轨迹相对完整且相似的 n 条轨迹，这 n 条轨迹代表的实际路线即为起始点和目的地之间选择最多的经验路线。

3.2.4 k-medoids 聚类提取质心轨迹

最大轨迹簇所代表的实际路线可以从中位数的思路出发，从上述 n 条轨迹中提取最能代表最大轨迹簇路线的一条轨迹。根据 3.2.2 中轨迹间距离的度量方法，对 n 条轨迹进行 k-medoids 聚类，将较大类的质心轨迹作为经验路线轨迹，因经过 3.2.3 节处理后的最大轨迹簇相似度较高，这里在进行 k-medoids 聚类时，类个数 k 设置不大于 2。

k-medoids 聚类与 k-means 聚类同属于划分聚类，且思路一致。k-means 算法是最普及的聚类算法，也是一个比较简单的聚类算法，算法接收一个未标记的数据集作为输入，根据预先输入的参数 k，将数据聚类成不同的 k 个簇，同时，k-means 算法也是一种无监督学习。k-medoids 聚类，在中心点选取时存在不一样的地方。k-means 聚类时，通过计算簇内所有点的平均值，当做这个簇的中心点。而在 k-medoids 聚类算法中，计算簇的中心点时，需要遍历这个簇的所有点，求出该点到簇内其他所有点的距离之和，将距离之和最小的点当做中心点。这两个聚类方法求中心点的差异，与求一组数据的平均值和中位数概念类似。k-means 聚类算法，适用于对数个圆形或类圆形分布数据进行聚类，虽然快速高效，但是也有一些明显的缺陷，如输入数据中存在极端的噪声，或随机初始点选择的不合理，最终结果会存在一些不合理的偏差。另外，k-means 聚类算法仅适用于输入数据可以计算一个平均值作为中心点的情况。相比较而言，k-medoids 聚类算法，由于选取的类似中位数的数据集中的具体点作为中心点，能有效的减少极端噪声点数据的影响。另外，在无法计算输入数据的平均点的情况下，只要能计算任意两个数据点的距

离的情况下，k-medoids 聚类算法依然能对输入数据进行聚类处理。

在现实情况中，浮动车车辆经过同样的路线或近似的路线所产生的轨迹，由于路况、车速等原因，轨迹点的数量差距很大，无法计算不同轨迹的平均轨迹。但是通过本论文 3.2.2 中定义的方法，可以计算两条轨迹间之间的距离，因此在求取质心轨迹的过程中，采用 k-medoids 算法对 DBSCAN 聚类后的轨迹进行聚类。提取的质心轨迹即为起始点与目的地间的经验路线轨迹，用质心轨迹代表经验路线再进行地图数据匹配，可以大量减少地图匹配的工作量。

3.3 轨迹与地图数据匹配

3.2 节中通过 DBSCAN 聚类，得到最大轨迹簇，计算每条轨迹在最大轨迹簇中的轨迹点与该轨迹总轨迹点数的比值，设置阈值提取经验轨迹；再对经验轨迹通过 k-medoids 聚类，提取质心经验轨迹。质心经验轨迹需要与地图数据进行匹配，生成地图上的道路序列。本节主要介绍质心轨迹与地图数据的匹配方法。

这里用到的矢量地图数据主要有 Link 表、Node 表、Link_shppoint 表，Link 表记录了道路 LINK 信息，Node 表记录了 LINK 起终点 node 信息，Link_shppoint 表记录了 LINK 上的形状点信息，数据说明如下。

表 3-1 地图数据说明-Link

Table 3-1 Field description of map data-Link

编号	字段名称	字段类型	字段描述
1	LINK_ID	int	link 编号
2	S_NODE_ID	int	node 编号，link 起点
3	E_NODE_ID	int	node 编号，link 终点
4	DIRECT	int	link 上的通行方向 1 双方向；2 顺方向；3 逆方向
5	LENGTH	double	link 长度，单位：米

表 3-2 地图数据说明-Link_shppoint

Table3-2 Field description of map data-Link_shppoint

编号	字段名称	字段类型	字段描述
1	LINK_ID	int	link 编号
2	X	double	link 上形状点坐标 X
3	Y	double	link 上形状点坐标 Y

表 3-3 地图数据说明-Node

Table3-3 Field description of map data-Node

编号	字段名称	字段类型	字段描述
1	NODE_ID	int	node 编号
2	X	double	node 点坐标 X
3	Y	double	node 点坐标 Y

3.3.1 地图匹配方法

轨迹匹配指的是行车 GPS 点位组成的轨迹与既有的地图道路数据进行匹配的过程，GPS 点位受各种因素影响存在精度问题，与地图上的道路数据吻合匹配情况并不理想。如果需要计算滴滴车辆的实际经过的道路，需要有合适的匹配算法，才能确保车辆的 GPS 点位与地图的道路数据匹配的准确性。为了减少 GPS 点位精度不准确带来的匹配问题，目前存在多种算法，实现 GPS 点位到道路的精确匹配。如基于几何匹配为基础的算法、基于拓扑关系的算法、基于概率统计的算法、基于模糊逻辑的地图匹配算法等。这些算法目前国内外的学者已经钻研的很深入。

基于几何关系的地图匹配算法，主要涉及点与点的匹配方法、点与曲线的匹配方法、曲线与曲线的匹配方法，前两种方法的优点是过程比较简单，计算速度快，但是匹配精度受道路密度、交叉口、高架桥等的影响，匹配结果很不理想。曲线与曲线的匹配，使用行车轨迹组成的曲线与道路曲线串进行匹配，选取距离最小的道路曲线串，精度比前两种方法有了很大的提高，但是计算量要高很多，总体精确度也受前面的 GPS 点位误差的影响。总的来说，这种匹配算法仅适用于简单道路地图的匹配，不适用于复杂的实际导航用的电子地图。

基于拓扑关系的算法，在基于几何的匹配算法的基础之上，增加了道路的拓扑关系，例如两条道路是立交还是平交、道路的通行方向是单向通行还是双向通行、道路的曲率等等，因此匹配准确率要比基于几何的道路匹配算法高很多。该算法分为 2 个过程：首先获取 GPS 轨迹点周围符合拓扑结构条件的备选道路，然后根据相应的匹配规则，筛选出匹配最好的道路，并计算 GPS 轨迹点到道路的投影位置。基于拓扑关系的算法，根据拓扑规则分为两种，简单拓扑关系匹配算法和加权匹配关系匹配算法。

基于概率统计的地图匹配算法，这种算法根据定义好的匹配规则，在指定的范围内，计算 GPS 轨迹点与道路匹配的概率，筛选出概率最高的道路并计算 GPS 轨迹点的投影点，获得车辆的位置。该算法的置信区间的范围，对算的执行速度

有很大影响。这种算法可以和基于拓扑的匹配算法结合，提高匹配的精确度。

基于模糊逻辑的算法，能从精确或模糊的信息中计算出准确结果，相对其他精确的算法或精确的方程来反应真实世界行为的匹配方法，模糊逻辑更适合与处理现实的不确定性、复杂性的问题。模糊逻辑匹配算法可以结合拓扑关系等道路的相关信息，更好的解决道路立交还是平交、主辅路区分等问题。

除了上述匹配算法，还有证据推理方法、卡尔曼滤波方法、先验知识方法、基于隐马尔科夫模型的匹配算法等。

3.3.2 球面距离与轨迹投影

轨迹点与地图匹配需计算轨迹点到周围道路的投影点及投影距离，这里对球面距离进行说明，并定义了点到道路的投影距离。

（1）球面距离

球面距离是地理学中最常用的距离计算方式，用于计算球面上两点间的最短距离。球面距离需要对球面几何就行求解，需要使用大量的三角函数进行计算。常用的计算公式有大圆公式和 Haversine 公式[40]，大圆距离的精度比 Haversine 要差，尤其在距离较大是误差也较大，本文进行地图匹配时采用 Haversine 距离公式来进行距离的计算。设球面上两点的经纬度坐标为 $A(x_1, y_1)$ 、$B(x_2, y_2)$，球面距离的计算方法如下：

1）经纬度坐标值为角度值，计算时需换算为弧度值，角度与弧度的换算为 $\pi/180\times$角度。

$$[x_1, y_1, x_2, y_2] = map(radians, [x_1, y_1, x_2, y_2]) \tag{3-13}$$

2）采用 Haversine 距离公式计算角距离 d。

$$d = 2\arcsin\sqrt{\sin^2\left(\frac{y_1 - y_2}{2}\right) + \cos y_1 \cos y_2 \sin^2\left(\frac{x_1 - x_2}{2}\right)} \tag{3-14}$$

3）利用弧长公式计算球面距离 $D_{hav}(A, B) = d * r$ ，$r = 6371393$ 为地球平均半径，单位为米。

（2）轨迹投影

轨迹点在地图道路线上投影点的计算是点到曲线的投影，因地图数据中道路线是含有形状点的曲线，将整条道路看作由形状点组成的线段的集合，点到曲线的投影距离可以转换为点到所有线段的投影的最短距离。轨迹点 s 和道路线 L，L 由起点 A 和终点 B 以及中间形状点 $a_1, \cdots\cdots, a_i$ 组成，计算轨迹点 s 在道路 L 上的

投影点的方法如下：

分别求 s 到线段 $Aa_1, a_1a_2, a_2a_3, a_3a_4, a_4a_5, a_5B$ 的最佳轨迹投影点，当垂直投影点在线段上时，则垂直投影点即为在该线段的最佳轨迹投影 s_i'，如图轨迹点 s 在线段 a_1a_2 的垂直投影 s_4' 在线段上，所以最佳轨迹投影为 s_4'；如果垂直投影点不在线段上，则该线段上离轨迹点最近的节点作为 s 在该线段的最佳轨迹投影，如图轨迹点 s 在线段 a_1a_2 的最佳轨迹投影 s_2' 为距离 s 最近的节点 a_2 。计算 s 到各线段最佳轨迹投影点 s_i' 的距离，这里使用的球面距离，距离最短的投影点 s' 即为 s 在道路 L 的投影点。轨迹点 s 到道路 L 的投影距离为：

$$D_{Proj}(s, L) = D_{hav}(s, s') \tag{3-15}$$

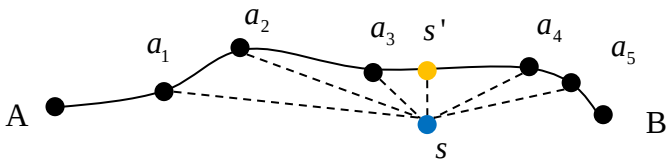


图 3-2 点到线的投影距离

Figure 3-2 The distance between a point and a line

3.3.3 基于改进的隐马可夫模型的地图匹配

在质心轨迹与道路地图数据的匹配过程中，我们采用了一种改进的隐马可夫模型（HMM）算法。隐马尔可夫模型是一个用来描述一个含有隐含未知参数的马尔可夫统计模型。给定一组观测变量，利用动态规划的思路根据当前状态的知识和信息求解条件概率最大的状态序列[41]。

浮动车轨迹数据前一个轨迹点和后一个轨迹点之间是连续的，HMM 进行地图匹配实质是用观察序列轨迹点序列来预测最大投影可能的实际道路序列，所以地图匹配的两个变量分别为：观测变量，即浮动车轨迹数据的位置信息（经度、纬度）；隐藏的状态变量，即浮动车轨迹的实际位置，对应地图上的道路线段。求解预测过程通过计算两个概率值进行度量，分别是观测概率和状态转移概率。

（1）观测概率的计算

观测概率：观测的浮动车轨迹点位离旁边路段上的距离越近，那么这个真实点在这个路段上的概率越大。一般认为，浮动车的轨迹点服从正态分布，本文中轨迹点 s 到周围道路 L_j 的观测概率的公式如下述公式 3-16。

$$P\left(s,L_j\right)=\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(D_{proj}(s,L_j)-\mu)^2}{2\sigma^2})\exp(-\frac{|\theta_1-\theta_2|}{180}) \quad (3\text{-}16)$$

其中，σ 表示 GPS 误差范围，结合浮动车 GPS 的实际情况，一般取值为 20m。μ 取值 0，$D_{Proj}(s,L_j)$ 为 s 点到周围道路 L_j 的投影距离。计算观察概率时，在距离的基础上增加方向权重 θ=θ_1-θ_2 ，轨迹点方向与候选道路方向夹角越小，投影到该道路的可能性越大。

本文在计算观测概率时，对轨迹点和地图数据进行网格化存储，增加索引，提高计算效率。观测概率是基于轨迹点到投影点的大圆距离计算获得的，这个距离越大概率越小。一般来说，电子地图中记录的道路的数量非常多，我们如果在计算每个轨迹点的观测概率过程中不对候选道路的数量或范围进行控制，那么计算出的观测概率会非常多，耗时也会很长。针对这种情况，本文对道路实验数据按照形状点的坐标进行网格划分，将数据划分成大小为横向 0.0003 经度差，纵向为 0.0003 维度差的网格，这些网格的尺寸边长大约 25 米，受所在经纬度影响存在很小的差异，不过不影响最终计算结果。

网格划分的处理：首先创建空的索引列表，用于存放网格编号、网格坐标范围（含左下角顶点坐标、右上角顶点坐标）、道路编号列表。遍历每一条道路上的所有形状点，根据划分原则计算归属的网格坐标范围。如果索引列表中包含该网格坐标范围，那么把该道路编号加入到对应的网格的道路编号列表；否则，增加一条索引记录，记录相应的网格编号、坐标范围、道路编号列表。

将所有的道路按照以上步骤处理，把划分后的结果存入缓存文件。在后续求解观测概率时，仅使用 GPS 点位所在网格和周边与该网格相邻的网格的道路数据。

（2）状态转移概率的计算

状态转移概率的计算有两种方法：前后两个轨迹点对应的真实位置点的距离越近，那么状态转移的概率越大；前后两个轨迹点对应的真实位置点的距离与前后两个轨迹点的距离差值越小，状态转移概率越大。本文选择第二种方式实现，对应的公式如下：

$$P(s_{k-1}^i,s_k^j)=\begin{cases}\exp(-\mu|D_{hav}(s_{k-1},s_k)-Dis(s_{k-1}^i,s_k^j)|)\mu,\ if\ \ L_i=L_j\\ \exp(-\mu|D_{hav}(s_{k-1},s_k)-Dis(s_{k-1}^i,s_k^j)|)\mu\times\rho,\ if\ \ L_i\neq L_j\end{cases} \quad (3\text{-}17)$$

状态转移概率表示的是 s_{k-1} 投影到上 L_i 的同时 s_k 投影到 L_j 的概率，$D_{hav}(s_{k-1},s_k)$ 为两个轨迹点的大圆距离，$Dis(s_{k-1}^i,s_k^j)$ 为轨迹投影到路段上的投影点在地图上的最短路网距离。在计算状态转移概率时增加权重，当相邻轨迹点的投影路段是不同道路时，乘系数 ρ，降低转移概率，可以降低分叉口投影 link 的错

误率，后续实验证明当 ρ 为 0.9 时，投影效果最佳。

备选投影点列表指每个浮动车轨迹点匹配电子地图中的道路时，在所有可能道路上的投影点列表，每个轨迹点都有自己的备选投影点列表。为了不遗漏合适的投影点且缩短整个匹配过程的计算时间，本文在计算每个轨迹点的备选投影点时，增加了判定条件，所有投影距离小于等于 20 米的备选投影点都会被选中，如果 20 米以内的备选投影点小于 5 个，那么继续增加投影道路的选择范围直到备选投影点数量达到 5 个。计算状态转移概率时，需要依次从 2 个邻接轨迹点的各自的备选投影点列表中各自取出 1 个点，计算所有取出的组合投影点之间的地图距离。

在计算投影点在地图上的最短路网距离 $Dis(s_{k-1}^{i}, s_{k}^{j})$ 时，为减少计算时间，在进行 HMM 匹配之前，使用 Dijkstra 算法，计算出所有的道路端点之间的最短距离，并存入缓存文本文件。Dijkstra 算法是一种贪心算法，是最短路径的经典算法之一，可以计算有向图或无向图的两个节点之间的最短路径。它是一种广度优先搜索的算法，计算时从起始点开始，向外逐层扩展，直到搜索到终点。HMM 匹配过程开始时，先从文本文件中读取数据，存入二维数组 short_dis，计算邻接轨迹点的投影点距离时，直接从 short_dis 数组中取出相应的值，大幅减少状态转移概率的计算时间。

表 3-4 short_dis 数组部分内容示例

Table 3-4 Sample of short_dis

	0	1	2	3	4	5	6
0	0	135.443	722.328	775.303	1849.011	1944.963	1750.593
1	135.443	0	586.885	639.86	1713.568	1809.52	1615.15
2	9999	586.885	0	52.975	1126.683	1222.635	1755.374
3	9999	639.86	52.975	0	1073.708	1169.66	1808.349
4	9999	1713.568	1126.683	1073.708	0	101.196	2497.238
5	9999	1809.52	1222.635	1169.66	101.196	0	2396.042
6	9999	1615.15	1755.374	1808.349	2497.238	2396.042	0

另外，计算投影点间最短距离时，考虑道路的通行方向。p、q 两个投影点在同一条道路时，p 到 q 的方向与道路通行方向相同时，计算这条道路上 p 到 q 的实际道路距离；如果 p 到 q 的方向与道路通行方向不同，则计算沿道路通行方向从 p 到道路端点 e 的距离、从数组 short_dis 中取出端点 e 到端点 s 的最短距离、计算沿道路通行方向从道路端点 s 到 q 点的距离，以上 3 个距离求和，即为 p 到 q 的投

影距离。p、q 两个投影点不在同一个道路时，计算沿道路通行方向 p 到道路端点的距离、沿道路通行方向从道路端点到投影点距离、从数组 short_dis 中取出相应道路端点的最短距离，求和即获得 p 到 q 投影点间的最短距离。如果投影点对应的道路是双向通行道路，那么要遍历所有可以通行的方向，求出相应的投影距离，并从中取出最小值，即为投影点在地图上的最短路网距离。

（3）维特比求解

维特比是一种求图的最短路径的算法，用动态规划的思路求解概率最大的路径。时间复杂度与序列长度成正比，与宽度的平方成正比，求解的过程速度比较快。

HMM 动态规划的求解思路如图 3-3，$s_{k-1}^{1'}$ 为 s_{k-1} 在候选投影道路 L_1 上的投影点，$s_{k-1}^{1'}$ 的初始投影概率为其观测概率，设 $s_{k-1}^{1'}$、$s_{k-1}^{2'}$ 的观测概率为 p_{k-1}^{1}、p_{k-1}^{2}，$s_k^{3'}$ 的投影概率依赖前一个轨迹点 s_{k-1} 各投影点的投影概率，设 $s_{k-1}^{1'}$、$s_{k-1}^{2'}$ 到 $s_k^{3'}$ 的转移概率分别为 q_1 和 q_2，$s_k^{3'}$ 的观测概率为 p_k^3，则 $s_k^{3'}$ 的投影概率为 $\max(p_{k-1}^{1} * q_1 * p_k^3, p_{k-1}^{2} * q_2 * p_k^3)$，并记录前一结点序号，依次计算至最后一个轨迹点所有投影点中最大投影概率的投影点，最后从后向前反推前一结点，得到所有投影点对应的道路序列 RD_list。

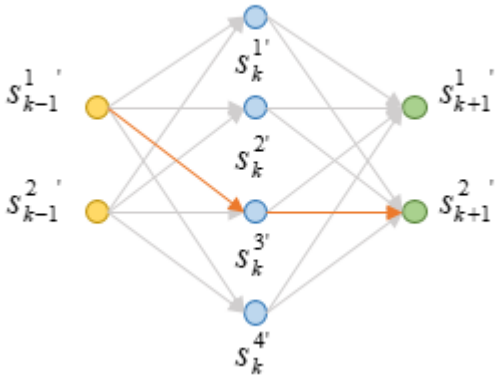


图 3-3 HMM 动态规划求最大概率

Figure 3-3 Maximum probability for HMM dynamic programming

3.4 经验路线提取

3.3 节基于改进的 HMM 地图匹配算法，将质心匹配到地图数据中，得到投影道路列表 RD_list，但 RD_list 中存在重复和不连贯的情况，需要去重后使用 A*算法补齐中断部分。

计算获取浮动车轨迹点的投影道路列表，主要会存在以下两种现象。第一个现象是每个轨迹点对应一条投影的道路，存在很多连续的轨迹点投影到同一个道

路上的情形，因此列表中存在很多重复的道路编号。需要对列表中重复的道路进行去重处理。第二个现象是两个连续的浮动车轨迹点，如果距离相对较远或刚好遇到路口，可能会出现这两个点投影的两条道路不连续的情况。这种情况需要将两条道路中间缺失的道路通过一定的计算方法补充上，本文采用 A*算法计算缺失道路。如图 3-4 所示，黑色点为质心轨迹点，红色为轨迹点对应的投影道路，蓝色框中存在明显的道路不连续。

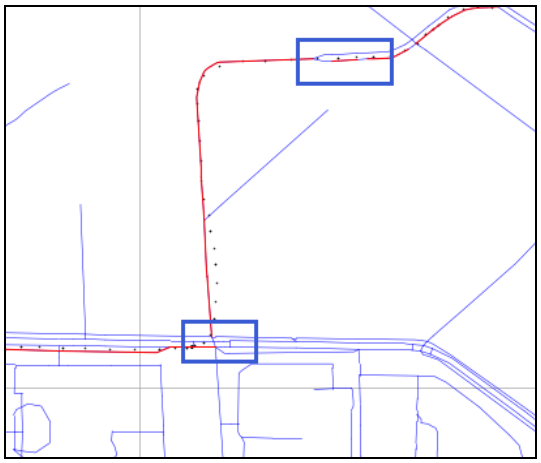


图 3-4 HMM 投影道路不连续情况

Figure 3-4 Projection path discontinuity of HMM

针对以上两种情况，对 RD_list 道路列表进行处理。首先遍历 RD_list 中的道路，判断当前道路与上一次取出的道路是否重复，如果重复则丢弃当前道路；如果不重复则继续判断当前道路的起点是否与上一次取出的道路的终点相同，如果不同则使用 A*算法计算上一次取出道路的终点与当前道路的起点之间的道路列表，将该道路列表和当前道路追加到结果列表，如果相同则继续遍历，直到最后一条道路循环结束后，结果列表的内容即为经验路径的道路编号。

3.5 实验与结果分析

3.5.1 实验数据

选择起点的坐标为（116.46700，40.01800），该位置在北京市朝阳区广顺桥附近；选择的终点坐标为（116.47600，40.04150），该位置在北京市朝阳区启明星双语学校附近，对起点和终点建立相应 buffer 区间，考虑建筑物或街区的水平面投影形状主要以方形或长方形为主，起点或终点的 buffer 区间分别设置为各自坐标为中心 100 米乘 100 米的正方形，对所有轨迹进行匹配，若一条轨迹的起点和终点

的坐标同时满足落到相应的起点和终点的 buffer 区间，那么这条轨迹被选中，存入到文件等待下一步的聚类处理。经过匹配计算，实验用的所有轨迹中共有 66 条轨迹符合匹配条件，相应的轨迹点共计 21116 个。如表 3-5 所示。

表 3-5 选定起终点后筛选的轨迹点

Figure 3-5 Selected trajectory point

No.	ID	X	Y
0	didi_1523873901_4a489……c21e7c	116.46689	40.01827
1	didi_1523873901_4a489……c21e7c	116.46689	40.01823
...	...	...	...
21115	didi_1524657189_1828df……b574a0	116.47603	40.04087

对轨迹点进行画图，黄色圈为起点范围，蓝色圈为终点范围。可以看到这些轨迹点组成的轨迹，从相同的起点位置到相同的目的地位置，浮动车车辆经过的路径，存在两个明显的现象。第一个现象：存在两种常走的路径，经过观测轨迹点分布图，选择左边路径的比例相对较小，选择右边路径的比例占相对较大。这些轨迹中，存在少量的浮动车车辆有绕路现象，结合浮动车的产品功能，推测浮动车车辆的乘客数量为 2 人或 2 人以上，浮动车车辆从起点接到第 1 位乘客，行进途中绕路接其他乘客，这些绕路的轨迹点，密度明显低于其他正常轨迹点，这些轨迹点可以视为噪声点。这 21116 个轨迹点的分布如图 3-5 所示。

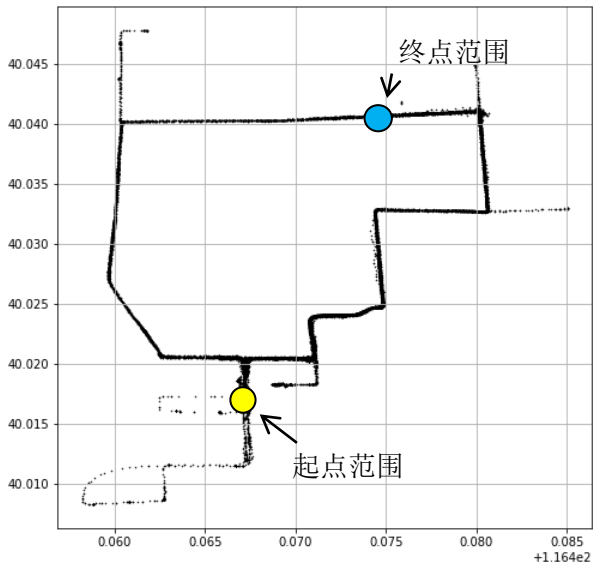


图 3-5 66 条选中轨迹的 12273 轨迹点

Figure 3-5 The 12273 points of 66 selected trajectories

3.5.2 实验结果

（1）轨迹聚类提取质心经验轨迹

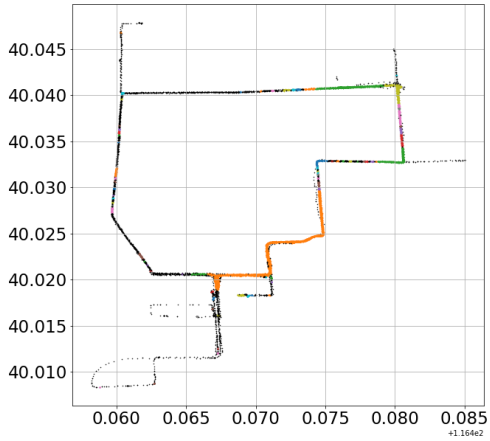
经验路线指的是浮动车车辆在同样起点同样重点时选择频率相对最高，且无绕路的路径，因此，有噪音点的绕路轨迹和选择相对较少的路径，都需要通过聚类方法进行筛除。对以上数据进行 DBSCAN 聚类测试，调整不同的ε半径参数、*MinPts* 邻域密度阈值，获取相应的聚类结果。本论文进行了多种参数组合，经过分析对比，选了其中的 4 组比较有代表性的参数进行展示。如图 3-6 所示，其中黑色点为聚类结果被标示为噪音点的轨迹点，其他不同颜色代表不同的聚类结果簇。

第一组参数ε半径参数为 0.00008、*MinPts* 邻域密度阈值为 20，聚类结果如图 3-6（a）所示。所有绕路的轨迹被标记为噪音点；左边少量浮动车车辆选择的普通路径，大部分被标记为噪音点。浮动车车辆常选的经验路径包含的轨迹点，被聚类为几个不同的簇，也有轨迹点被标记为噪音点。本组参数可以把浮动车车辆选择较少的路径和浮动车车辆绕路的路径筛除掉。

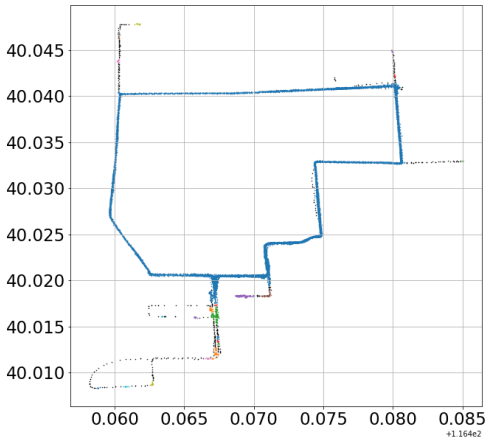
第二组参数ε半径参数为 0.00015、*MinPts* 邻域密度阈值为 10，聚类结果如图 3-6（b）所示。左右两条路径被聚类到同一个簇中，如图中蓝色轨迹点。绕路的轨迹点被聚类到噪音簇。本组参数无法支持选择提取经验路径的目的。

第三组参数ε半径参数为 0.00015、*MinPts* 邻域密度阈值为 20，聚类结果如图 3-6（c）所示。浮动车车辆常选的路径被聚类到同一个簇中，如图中绿色轨迹点所示；绕路的轨迹点被标记为噪声点；浮动车车辆少选的路径被聚类到不同的簇中。

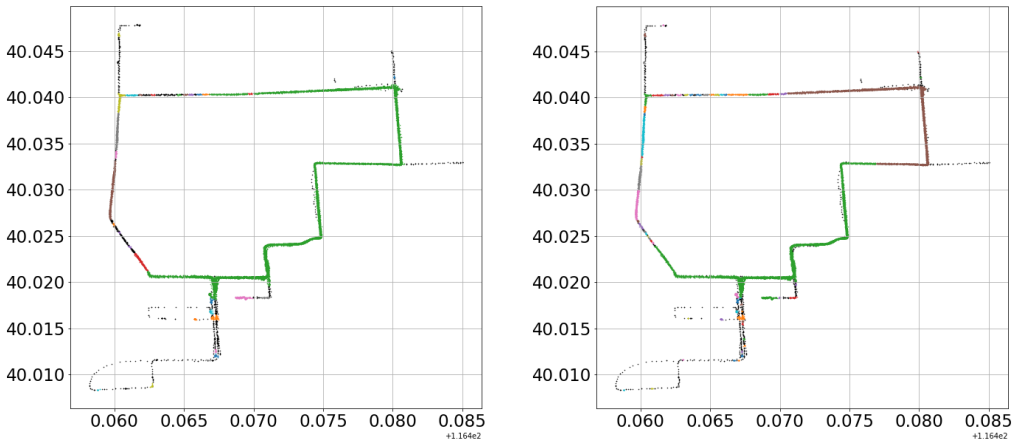
第四组ε半径参数为 0.00010、*MinPts* 邻域密度阈值为 10，聚类结果如图 3-6（d）所示。常选和不常选的路径都被聚类到不同的簇中，不支持选择提取经验路径的目的。



(a) ε:0.00008, *MinPts*:20



(b) ε:0.00015, *MinPts*:10



(c) ε:0.00015, *MinPts*:20 (d) ε:0.00010, *MinPts*:10

图 3-6 DBSCAN 聚类结果

Figure 3-6 The result of DBSCAN

经过对以上不同参数 DBSCAN 聚类结果的对比分析，在ε半径参数、*MinPts*邻域密度阈值参数设置，可以得到不同的聚类分簇。第一组和第三组参数都可以支持区分经验路径和其他路径，但是选择这两组不同的参数后续的处理方法不同，选择第一组的参数后续处理思路为筛除掉被标记为噪声点超过一定比例的轨迹，剩余的轨迹为经验路径轨迹。第三组参数的处理方法为，选出占比最大的簇，计算每条轨迹里被分到这个簇的比例，选择这个比例超过相应阈值的轨迹。两种方法都可以获得最终的轨迹，本论文选择后一种思路进行经验路径的提取。

对选定参数后的 DBSCAN 聚类分簇结果进行各簇轨迹点占比统计，每个簇的占比如表 3-6 所示。浮动车车辆选择最多的路径点，被聚类到簇 2，该簇的轨迹点占轨迹点总量的 78.62%。

表 3-6 DBSCAN 聚类各簇的轨迹点占比

Table 3-6 Percentage of point in each cluster

分簇	占比
簇 0 的占比	0.60%
簇 1 的占比	0.50%
簇 2 的占比	78.62%
簇 3 的占比	1.39%
簇 4 的占比	0.12%
...	...
簇 31 的占比	0.09%

定义轨迹最大簇轨迹点的占比为单条轨迹中包含簇 2 轨迹点的个数与该轨迹总轨迹点数量的比值，对每条轨迹进行统计，统计结果如表 3-7 所示。其中占比超过 60%的轨迹数量为 41 条，小于 60%的轨迹数量为 25 条。

表 3-7 每条轨迹的最大簇轨迹点的占比

Table 3-7 Percentage of point of the largest cluster in each trajectory

轨迹 ID	占比
didi_1524060515_4ad52c8fc43607273519d4932185a8d0	22.80%
didi_1524396333_40e5369e45cf2353c46b23308e37dff5	35.62%
didi_1524145704_19936f98e621953dc8af3275a90d6843	36.09%
didi_1524317730_1bc0ff5decdc32bb0e10e14f30bc84ab	37.12%
didi_1524582024_9d019215ce6eefe338afac5f1c598f51	38.46%
...	...
didi_1524458627_7d7bfe0d58625fabd32a75f7550f5c88	100.00%
didi_1524657189_1828df77fce1d479fc188dda25b574a0	100.00%
didi_1524315827_e71b4547321bec7d7a8a145f4b28c34d	100.00%
didi_1524148986_e752723baa7c5ab6a588b0cd4cdcc640	100.00%

选择占比阈值为 60%时，可以将所有轨迹区分成浮动车车辆常走路线即经验路线和走的较少的路线或有绕路的两种情况。占比超过 60%的浮动车轨迹点如图 3-7 所示：

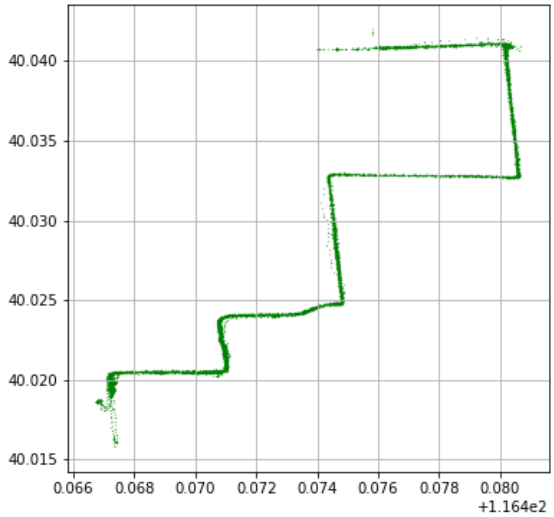


图 3-7 DBSCAN 聚类的经验路径轨迹点

Figure 3-7 Result of DBSCAN

经过 DBSCAN 聚类后的轨迹数据，已经屏蔽掉选择偏少的轨迹和有绕路的噪

声轨迹。这些聚类后的数据，相对比较集中，因此后续 k-medoids 聚类过程，将轨迹数据聚类为 2 簇，即 k 设置为 2。采用前面章节 3.2.2 里定义的轨迹距离计算公式，对任意两个轨迹进行距离计算，并将计算结果存入二维数组。将距离二维数组和 k 作为输入，进行 k-medoids 聚类计算，选出聚类后较大的簇，其质心轨迹点作为经验路线的轨迹点，质心轨迹点组成的线串如下图 3-8 ，其中蓝色为轨迹点，红色为质心轨迹连线。

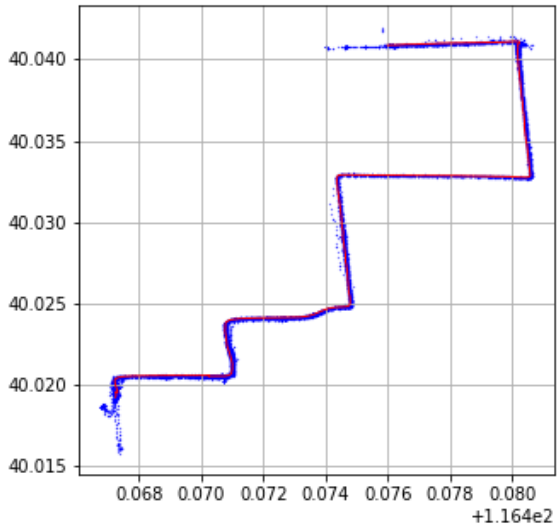


图 3-8 k-medoids 聚类结果

Figure 3-8 Result of k-medoids

（2）HMM 地图匹配

上述实验中提取的整条质心轨迹有 299 个轨迹点，对轨迹聚类生成的质心经验轨迹进行地图匹配。基于改进的 HMM 算法进行地图匹配，299 中有 3 个点投影异常，匹配正确性在 99%，匹配结果如图 3-11 所示。

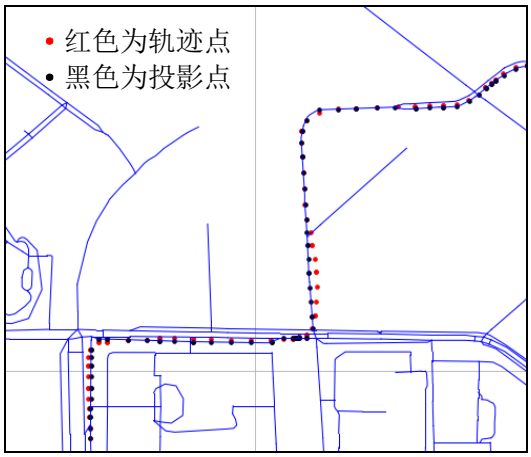


图 3-9 HMM 投影结果

Figure 3-9 Result of HMM projection

（3）经验路线提取

对 HMM 投影点所对应的道路列表 RD_list，进行去重后使用 A*算法补齐中断部分，得到地图上完整且连通的道路 link 序列，共 65 条 link，如图 3-12 所示。

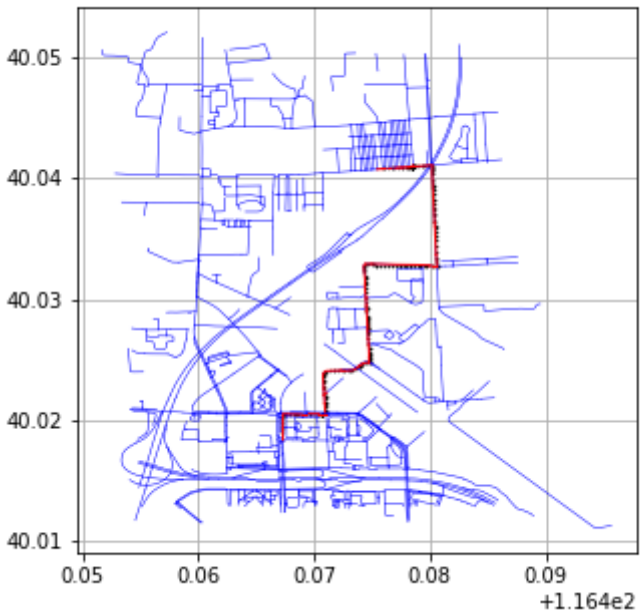


图 3-10 经验路线提取结果

Figure 3-10 Result of HMM for Experienced route

3.5.3 算法优化效果

实验过程中发现使用普通的状态转移概率公式，进行浮动车轨迹与道路进行匹配，计算出来的匹配道路，在十字路口或丁字路口是双向通行道路时，有很小的概率出现不合理的现象。如图 3-9 所示，圆圈中的黑色轨迹点，计算出来的投影道路是丁字路口的横向道路，但是结合这组轨迹点的整体情况，这个轨迹点投影到左边的竖向道路更为合理。在使用未做改善的状态转移概率算法时，对比一下这个轨迹点投影到横向和竖向 2 条道路的观测概率和状态转移概率，这个轨迹点的投影到横向道路时轨迹点与投影点距离接近 0，观测概率达到最大值，要大于投影到左边竖向道路的情况，而状态转移概率相差不大，因此最终匹配结果选择了不合理的横向道路。这个现象可以概括为连续滴滴轨迹点连续的投影到同一条道路在路口偶发投影到另一个道路上，本文针对这个现象对状态转移概率进行了改善，如果两个点投影到不同道路上，那么投影概率在默认投影概率的公式基础上乘以 0 到 1 之间的一个系数；如果投影到相同道路上，投影概率公式不做变化。经过反复实验，这个系数选为 0.9 时，可以解决类似情况而不带来其他问题。

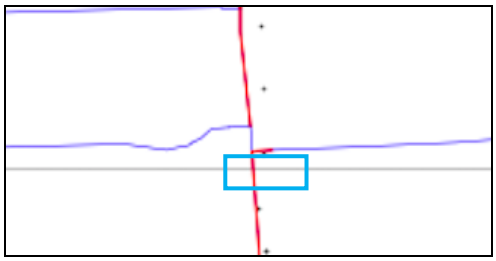


图 3-11 改善前投影道路

Figure 3-11 Unimproved projection

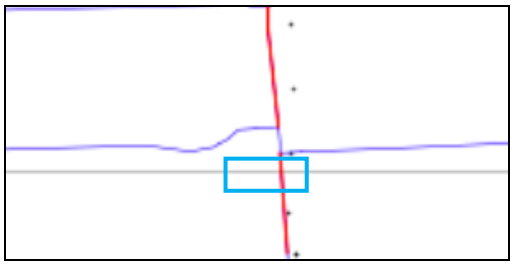


图 3-12 改善后投影道路

Figure 3-12 Projection after improvement

HMM 进行地图匹配，在匹配准确性方面，除上面对状态转移概率进行优化外，还包括之前提到的计算观察概率时，在距离的基础上增加方向权重$\theta=\theta_1-\theta_2$，来提高匹配的准确性。分别对单个改善和联合改善效果的匹配准确性进行统计，如表 3-8。

表 3-8 地图匹配准确率

Table 3-8 Accuracy of Map Matching

匹配方法	匹配准确率(%)
改善前 HMM 匹配	89
观测概率改善后	98
状态转移概率改善后	95
两种改善叠加	99

HMM 算法另一方面的优化主要体现在概率计算的效率上，通过网格化存储，只计算轨迹点到当前网格和周边网格内 link 的观测概率，可以减少 90%的计算量；对 node 间最短距离的缓存，缓存前后转移概率中 node 最短距离计算耗时如表 3-9。通过以上改善能大幅度提高观测概率和状态转移概率的计算效率。

表 3-9 node 最短距离计算效率

Table 3-9 Computing efficiency of shortest distance

node 数量（个）	100	500	1000	1500	2000
改善前计算耗时（秒）	1.52	3.51	6.93	9.87	13.7
改善后计算耗时（秒）	0.9	0.99	1.01	1.02	1.03

3.6 本章小结

本章首先介绍了基于浮动车轨迹数据的经验路线提取的整体思路和技术流

程。其次对传统轨迹聚类和地图匹配的方法进行了介绍，在此基础上详细描述了本文中算法的创新和经验路线提取方法的实现。经验路线提取的主要研究如下：

（1）经验路线提取，基于 DBSCAN 和 k-medoids 的轨迹聚类，有效识别选择最多的线路，并过滤掉选择偏少的分支轨迹和有绕路的噪声轨迹，提出一种轨迹间距离度量的方法，根据轨迹间相似度聚类提取质心轨迹。然后对质心轨迹进行地图匹配，生成经验路线。

（2）用改进的 HMM 算法进行地图匹配，算法创新点有：

1）对轨迹点和地图数据进行网格化存储，增加索引，提高计算效率；计算观测概率时根据网格限定计算范围，减少计算量。

2）计算地图上任意两个 node 点的最短距离进行缓存，大幅减少状态转移概率的计算时间。

3）设置两个系数，增加投影的准确性，计算观测概率部分增加轨迹点方向和道路方向差异量的系数θ，差异量越小，观测概率越大；计算转移概率时，当相邻轨迹点投影到不同道路上时，通过控制系数ρ，减小状态转移概率值，降低路口处轨迹点投影错误的概率。

4 基于经验路线的路径规划

第三章中通过对轨迹数据进行聚类和地图匹配，提取到含有浮动车轨迹经验的路线，本章主要研究经验路线在路径规划中的应用。参考第二章中路径长度和轨迹分布，提取热点 OD 间的经验路线，生成经验路线图层，并建立与基础路网的拓扑关系。采用分层的路径规划方法，结合常规路径规划算法 A*算法，提出基于经验路线的分层路径规划方法，将浮动车轨迹的经验加入到路径规划中，提供更具经验价值的路径选择，缩短起点与目的地之间的通行时长或路径长度，提升出行体验。

4.1 A*算法

本文的研究，主要是基于海量浮动车轨迹抽取滴滴司机日常行车的经验路径，采用的路径规划算法是在静态路径规划方法基础上增加了经验路径图层，对实际路径计算有一定的参考意义。在 3.4 节进行经验路线提取时为补全缺失道路，采用基础的 A*算法进行了路径补全。本章在进行静态路径规划时，依然采用 A*算法，不同于前面的路径补全，本章在使用 A*算法进行分层路径规划时，A*的启发函数需充分考虑路网特征，包括道路等级、交通限制、平均行驶速度等，以满足路径整体的规划需求。

A*算法是一种应用范围比较广的计算最短路径的算法，在一些电子地图进行计算路径规划、电子游戏进行玩家或怪物行走路径计算时都有应用，与贪婪算法不一样，贪婪算法适合动态规划，寻找局部最优解，不保证计算结果是最优解。A*算法是静态网格中求解最短路最有效的方法。计算的过程比较耗时，不适合在寻路频率较高的场景下使用。一般来说 A*算法适合对计算结果的精度要求较高的情形。与启发式的搜索一样，能够根据改变网格密度、网格耗散来进行调整精确度。在策略游戏的策略搜索或方块格子游戏中的格子寻路这类场景应用比较合适。

A*算法在开始计算前，不需要对地图数据进行做任何处理，是一种直接算法。同时 A*算法也是启发式算法，使用了启发函数，启发函数一般记为 *h(n)*，表示从节点 *n* 到目的地节点的估计值。起点节点到节点 *n* 的实际距离称为耗散函数，一般记为 *g(n)*。*n* 节点的评估函数 $f(n)=g(n)+h(n)$，表示耗散函数与启发函数的和。在持续寻路的过程中，A*算法根据评估函数 *f(n)*的值，找到一个合适的节点，然后展开该节点，避免了每次都需要展开所有节点的情况，极大的提高了寻路的效率。A*算法的的核心是设置一个合适的评估函数，评估函数的构造对搜索结果有决定

性作用。对节点 *n* 来说 *g(n)*的值基本是固定的，因此启发函数 *h(n)*的选择非常重要，它能控制 A*算法计算过程中的行为。如果把 *h(n)*设置为 0，这时候 *f(n)=g(n)*，A*算法变成了 Dijkstra 算法，也能计算出目标路径。启发函数越小，在 A*寻路时需要探索的节点越多，寻路过程就会变慢。启发函数如果能与节点 *n* 到目的地距离的实际情况一致，那么整个寻路过程探索的节点就是最终路径上的节点，寻路过程达到最快的情况。启发函数如果设置的比 *g(n)*大非常多，相当于只有启发函数起作用，A*算法接近 BFS 算法。

常用的启发函数 *h(n)*二维坐标下有以下几个距离度量方法：

（1）曼哈顿距离，表示在标准坐标系上的绝对坐标轴距之和，$A(x_1, y_1)$、$B(x_2, y_2)$两点的距离公式如下：

$$d(A,B) = |x_1 - x_2| + |y_1 - y_2| \tag{4-1}$$

（2）欧几里得距离，也叫欧式距离，表示两点之间的直线距离。$A(x_1, y_1)$、$B(x_2, y_2)$两点的距离公式如下：

$$d(A,B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4-2}$$

（3）平方后的欧几里得距离，表示两点之间直线距离的平方，在计算过程比欧几里得距离少一步求开方结果的过程，因此效率相对高一些，不过存在单位不一致的影响。$A(x_1, y_1)$、$B(x_2, y_2)$两点的距离公式如下：

$$d(A,B) = (x_1 - x_2)^2 + (y_1 - y_2)^2 \tag{4-3}$$

（4）另外还有各坐标数值差的最大值作为距离的切比雪夫距离等，$A(x_1, y_1)$、$B(x_2, y_2)$两点的距离公式为：

$$d(A,B) = \max(|x_1 - x_2|, |y_1 - y_2|) \tag{4-4}$$

本文实验用的电子地图数据，道路属性里的长度字段记录的是以米为单位的实际长度，这个长度即是道路两个端点之间的距离。如果我们在 A*算法寻路过程中，直接使用以上几个公式之一作为启发函数，那么 *h(n)*的值远小于实际情况，会导致 A*算法展开的点过多，影响整个计算过程的效率。为了使启发函数 *h(n)*的结果与实际情况更接近，本文采用的是两点间的大圆距离公式作为启发函数，这时评估函数中 *g(n)*和 *h(n)*两者的单位是统一的，寻路结果的准确性和寻路的速度都得到了保障。

4.2 基于经验路线的分层路径规划

传统的经验路线计算方法，一般是在浮动车轨迹数据中抽取司机的经验知识，参考这些轨迹数据中的行车速度、计算访问频次较高的道路，获得司机频繁选取的优选道路路段，将这些道路构建一个经验图层，并设置相应的权值，供用户计算通行路线时使用。传统的经验路线主要考虑的出租车司机热选的单独路段，并没有考虑从指定起点到目的地范围出租车司机选取整条轨迹的经验。本文提出一种构造包含起点、目的地、整条轨迹作为经验路线三个要素的经验路线图层的方法。

根据前面章节的计算方法，对所有浮动车轨迹进行计算处理，抽取经验路线，建立单独的经验路线图层，数据存放在库中，数据结构如表 4-1 与表 4-2。

表 4-1 经验路线数据格式

Table 4-1 Field description of experienced route

编号	字段名称	字段类型	字段描述
1	GROUP_ID	int	路线编号
2	S_NODE_X	double	路线起点坐标 X
3	S_NODE_Y	double	路线起点坐标 Y
4	E_NODE_X	double	路线终点坐标 X
5	E_NODE_Y	double	路线终点坐标 Y

表 4-2 经验路线组成 LINK

Table 4-2 Composition LINK of experienced route

编号	字段名称	字段类型	字段描述
1	GROUP_ID	int	路线编号
2	LINK_ID	int	组成 LINK 号码
3	SEQ_NUM	int	LINK 在经验路线中的序号
4	DIRECT	int	LINK 方向

在用户进行实际路径计算时，对起点和目的地坐标设置缓冲区范围，与经验路线起终点坐标进行匹配，分为 3 种情况：起点、目的地缓冲区范围刚好匹配到经验路线的起终点坐标，直接采用经验路线进行路径计算；起点、目的地缓冲区匹配不到经验路线的起终点坐标，但起点到目的地覆盖范围包含完整经验路线时，分别计算经验路线、A*算法路径，匹配度较高，使用经验路线，匹配度一般，使用 A*算法路径；起点、目的地起点缓冲区匹配不到经验路线的起终点坐标，且起点到目的地覆盖范围不包含完整经验路线时，直接采用 A*算法路径。

基于经验路线的路径计算方法的流程如图 4-1 所示。

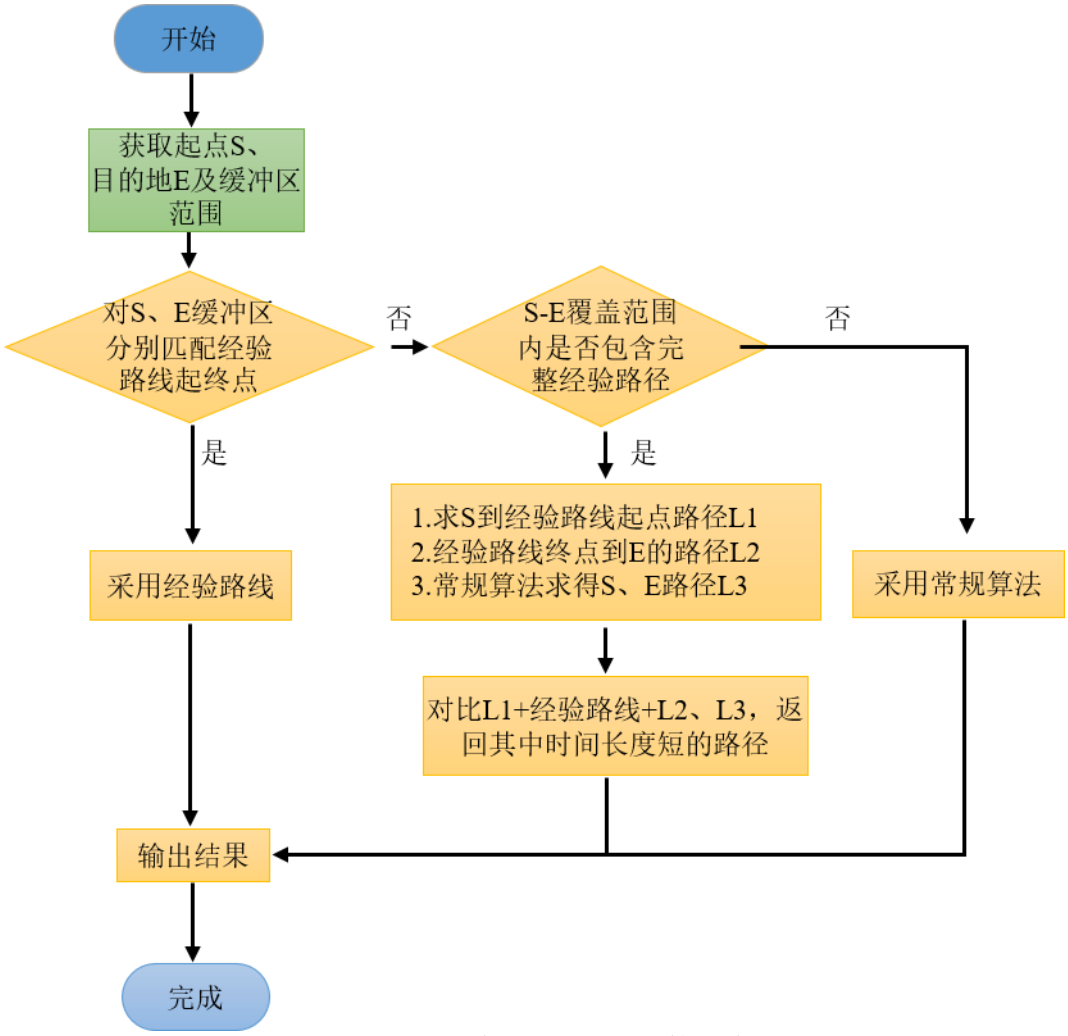


图 4-1 基于经验路线图层的路径计算方法

Figure 4-1 Path calculation method based on experience route layer

4.3 实验与结果分析

4.3.1 实验方案

本文中使用的数据都是真实数据，浮动车轨迹数据采用的是滴滴出租车、快车等运营车辆的脱敏轨迹数据，电子地图的道路数据采用的是北京市区域的数据。

滴滴运营车辆的司机常年在路上开车，对拥堵情况、限速、红绿灯等待时长等方面有丰富的经验，大部分情况在选取路径时采用通行时间较短或距离较短的道路，因此本实验在路径的长度和通行时长两个维度进行对比。本实验对比参照的对象，选取了 B2C 占用率较高的百度地图给出的最短路径、推荐路径。路径长度对比时，经验路径的长度选取经验轨迹匹配的道路列表的长度；时长对比时，

经验路径的通行时长采用滴滴轨迹终点与起点的时间字段的差值；对比参考值采用百度地图提供的距离长度和预估时长。

从滴滴轨迹中选择 20 个起点目的地组对，按照最短距离从小到大进行排序，并对每一个组对编写唯一 ID，在后续实验对比结果画图时作为横坐标使用。详细信息见表 4-3。

表 4-3 OD 组对的选取

Table 4-3 Selected OD pair

NO.	起点（O）	终点（D）	最短距离	OD 组对 ID
1	中国人民大学	北京理工大学主校区	2962	zgrd-bjlg
2	广顺北大街	京东顺白路物流点	4313	gsbd-jdsb
3	地坛北里小区北门	比如世界购物中心	5057	dtbl-brsj
4	凤凰置地广场-A 座	北京朝阳医院	5582	fhzd-cyyy
5	国家行政学院	车公庄地铁站	7168	xzxy-cgzz
6	卧龙小区	太阳宫公园	7621	wlxq-tygy
7	卧龙小区	天虹商场(国展店)	8326	wlxq-thgz
8	西直门地铁站	北京西站停车场	9516	xzmz-bjxz
9	凯迪拉克中心	冠英园小区	10113	kdlk-gyyx
10	苏州桥	西什库天主教堂	11479	szq-xskj
11	西直门地铁站	北京南站	12073	xzmz-bjnz
12	八家嘉园	中关村壹号	13278	baja-zgcy
13	北街家园	中关村壹号	15405	bjjy-zgcy
14	苏州桥	八大胡同	15795	szq-bdht
15	国际财经中心-A 座	凯德 MALL(太阳宫店)	16098	gjcj-kdty
16	华严北里东区	北京西站停车场	16283	hybl-bjxz
17	凤凰置地广场-A 座	北京首都国际机场	19861	fhzd-bjjc
18	用友科技园西区北门	地坛北里小区北门	24457	yyxq-dtbl
19	博泰嘉华大厦	IC-PARK 停车场	25618	btjh-icpk
20	八家嘉园	北京首都国际机场	31296	baja-bjjc

4.3.2 实验结果分析

（1）路径长度对比分析

图 4-2 为路径长度的对比结果，纵轴为路径的距离长度，单位为米。蓝色为百

度地图最短路径，橙色为百度地图推荐路径，灰色为经验路径。从柱状图中可以明显看出，经验路径的距离，有超过 70%的比例，介于百度地图的最短路径和推荐路径之间，经验路径的距离略超推荐路径的情况不到 30%。

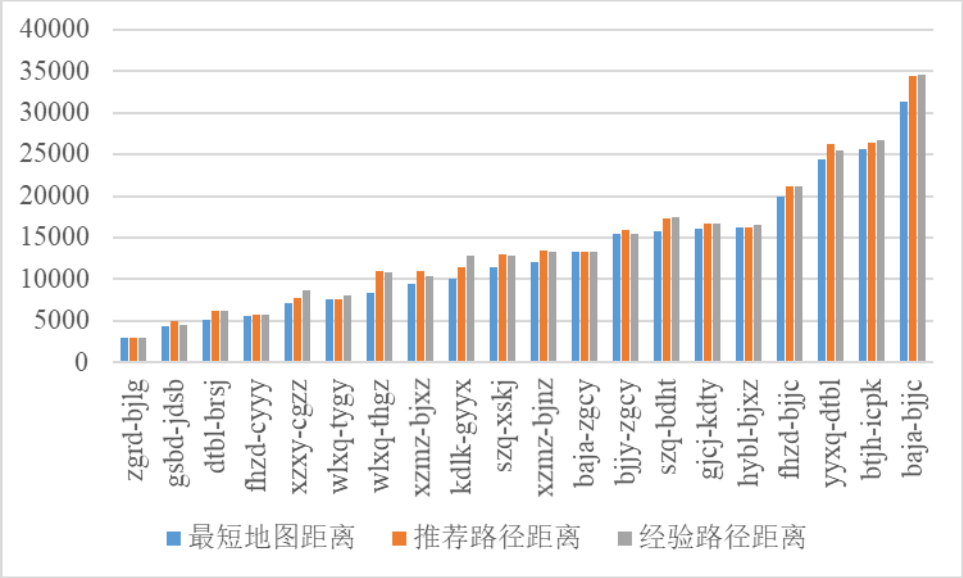


图 4-2 路径距离对比图

Figure 4-2 Path length between O and D

（2）路径通行时长对比分析

图 4-3 为路径通行时长的对比图，纵轴为路径的时长，单位为分钟。从柱状图中可以明显看出，经验路径的通行时长，在总路径时长小于 15 分钟时，超过百度推荐路径时长的比例达到 50%，推测乘客上下车的时间在经验路径的轨迹中占用了一定时长，不影响前面章节 k-medoids 聚类抽出的质心轨迹作为经验路径。在总时长超过 20 分钟的情况下，经验路线时长略小于百度最短路径和推荐路径的比例达到 80%，推测滴滴司机的寻路经验和驾驶技巧发挥了作用。

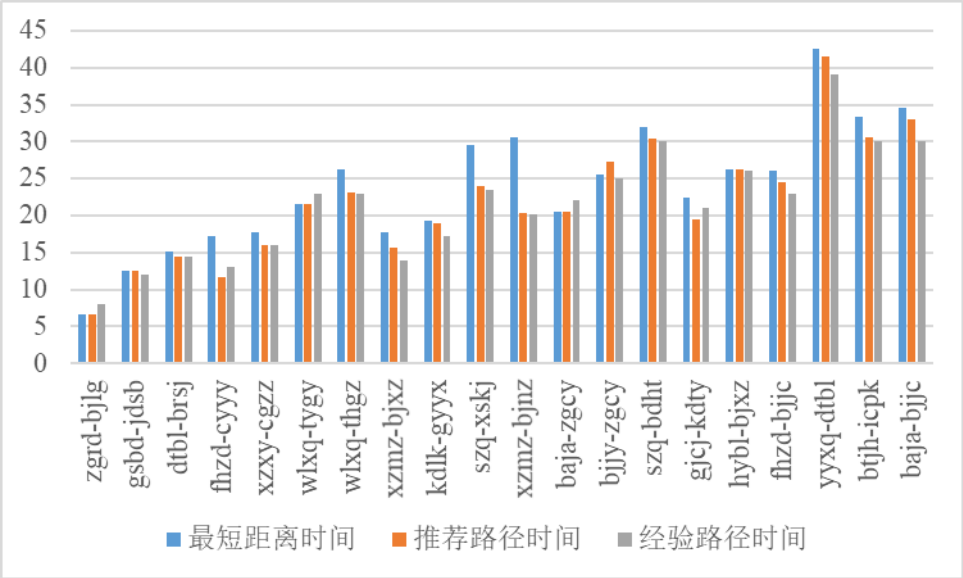


图 4-3 OD 路径通行时长

Figure 4-3 Path time cost between O and D

4.4 本章小结

本章采用分层的路径规划，结合传统 A*算法，将提取好的经验路线作为单独图层加入到路径计算中，实现基于经验路线的路径规划算法。实验结果的验证通过选取北京市 20 组 OD 的路径长度和平均行驶时间与百度地图的实际搜索结果进行对比分析，整体上基于经验路线的呈现较好优势。

5 总结与展望

5.1 总结

浮动车轨迹一定程度上反映着城市道路的行驶规律，蕴含着司机的驾驶经验智慧，综合考虑了道路等级、红绿灯数量、拥堵程度及周围环境等各种因素。将浮动车轨迹的经验加入到路径规划中，提供更具经验价值的路径选择，缩短起点与目的地之间的通行时长或路径长度，提升出行体验。

本文主要从浮动车轨迹数据本身出发，以数据为驱动，对轨迹的时空特征和轨迹行为特征进行分析，了解人们的出行规律及出行特征，提出基于轨迹行为特征分类的模式匹配方法进行交通信息识别。通过轨迹聚类和地图匹配，提取出最大程度上包含司机典型经验的完整路线，在此基础上，实现基于经验路线的分层路径规划。

本文的主要研究内容和结论如下：

（1）浮动车轨迹数据的经验路线提取。针对轨迹聚类，本文提出一种轨迹间距离度量的方法并根据轨迹间相似度聚类提取质心经验轨迹。基于 DBSCAN 和 k-medoids 的轨迹聚类，有效识别选择最多的线路，并过滤掉选择偏少的分支轨迹和有绕路的噪声轨迹。然后对质心经验轨迹进行地图匹配，去重并补全经验路线的道路序列，得到地图上完整且连通的道路序列。质心经验轨迹的提取为轨迹与地图的匹配节省的大量的工作。

（2）使用改进的 HMM 算法进行地图匹配。针对地图匹配的准确性和投影情况的多变性，提出改善的观测概率计算方法和转移概率计算方法，来降低轨迹点的错误投影概率。针对地图匹配的计算效率，提出网格化的存储和基于网格范围的计算来提高观测概率的计算效率，并通过计算并缓存地图上任意两点的最短距离，大幅度减少状态转移概率的计算量。

（3）基于经验路线的路径规划研究。针对路径规划方法的研究，提出基于经验路线图层的分层路径规划方法。对北京市热点 OD 进行经验路线提取，存储为单独的经验路线图层，采用分层的路径规划算法，结合传统的 A*算法，将经验路线图层融入到路径规划，形成一种基于经验路线的分层路径规划方法。通过实验对比，基于经验路线的路径规划结果从时间和距离上呈现一定优势。

5.2 展望

本文基于浮动车轨迹数据实现基于经验路线的路径规划，但因个人精力和能力的原因一些内容没有研究到位，值得进一步研究和探索，主要有以下几个方面：

（1）随着 GPS 精度提升，基于浮动车轨迹数据的行为特征分析可以实现更多更详细的交通信息挖掘，例如车道级的车道位置和车道宽度等交通信息。车道级的路网结构和交通信息是高精度地图重要的数据基础，可以为自动驾驶提供准确的道路信息。

（2）本文研究的经验路线方法是基于历史数据的静态路径规划，后续研究可以引入分布式计算，运用大数据平台对轨迹数据进行实时分析与挖掘，将挖掘结果实时加入到常规路径规划算法，实现基于轨迹挖掘的动态路径规划。

（3）浮动车轨迹数据量呈现爆发式增长，数据存储、分析手段、计算性能等成为目前面临的亟待解决的问题。提高分析挖掘的速度、高度集成挖掘过程算法、增加可视化的挖掘过程，更便于被理解和接受并应用于业务领域。

参考文献

[1] 郑珂,朱敦尧.浮动车技术应用研究进展[J].现代电子技术,2016,39(11):156-160.
[2] 黄美娴. 基于浮动车数据(FCD)的道路实时速度匹配与数据挖掘[D]. 同济大学, 2009.
[3] Bernstein D , Kornhauser A . An introduction to map matching for personal navigation assistants[J]. Geometric Distributions, 1998, 122(7):1082-1083.
[4] White C E , Bernstein D , Kornhauser A L . Some map matching algorithms for personal navigation assistants[J]. Transportation Research Part C, 2000, 8(1-6):91-108.
[5] Phuyal B P . Method and use of aggregated dead reckoning sensor and GPS data for map matching[C]. Proceedings of The Institute of Navigation. Oregon Convention Center, Portland, OR. 2002, 430-437
[6] Greenfeld, Josh. Matching GPS observations to locations on a digital map[C].//81st Annual Meeting of Transportation Research Board. 2002, 1(3):164-173.
[7] Noland R B, Quddus M, Ochieng W Y. Map-matching in complex urban road networks[J]. Revista Brasileira de Cartografia, 2003, 55(2):1-14.
[8] Newson P, Krumm J. Hidden Markov map matching through noise and sparseness[C]. Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2009:336-343.
[9] 吴华意，黄蕊，游兰，等. 出租车轨迹数据挖掘进展[J]. 测绘学报，2019，48(11): 1341-1356.
[10] Tang K , Chen S , Liu Z . Citywide Spatial-Temporal Travel Time Estimation Using Big and Sparse Trajectories[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(12):4023-4034.
[11] 吴佩莉, 刘奎恩, 郝身刚, 等. 基于浮动车数据的快速交通拥堵监控[J]. 计算机研究与发展, 2014, 51(01):189-198.
[12] Zhao S , Zhao P , Cui Y . A network centrality measure framework for analyzing urban traffic flow: A case study of Wuhan, China[J].Physica A: Statistical Mechanics and its Applications, 2017(478): 143-157.
[13] Chen, Q, Song, X, Yamada, H, et al. Learning deep representation from big and heterogeneous data for traffic accident inference[C].//AAAI Conference on Artificial Intelligence. AAAI, 2016:338–344.
[14] Agamennoni G , Nieto J I , Nebot E M . Robust inference of principal road paths for intelligent transportation systems[J]. IEEE Transactions on Intelligent Transportation Systems, 2011, 12(1): 298-308.
[15] Chen C , Cheng Y . Roads digital map generation with multi-track GPS data[C].// International Workshop on International Workshop on Education Technology & Training. IEEE, 2008:508-511.
[16] Yao H , Wu F , Ke J , et al. Deep multi-view spatial-temporal network for taxi demand pediction[C].//Proceeding of the 32nd AAAI Conference on Artifical Intelligence. AAAI, 2018:2588-2595.

[17] 张健钦, 仇培元, 杜明义. 基于时空轨迹数据的出行特征挖掘方法[J]. 交通运输系统工程与信息, 2014, 14(06):72-78.
[18] Dijkstra E W. A note on two problems in connexion with graphs[J]. Numerische mathematic,1959,1(1):269-271
[19] Cheng, C, Riley, R, Kumar, S. P. R, et al. A loop-free extended Bellman-Ford routing protocol without bouncing effect[C].//ACM SIGCOMM Computer Communication Review. ACM, 1989, 19(4):224-236.
[20] Hart P E , Nilsson N J , Raphael B . A formal basis for the heuristic determination of minimum cost paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2):100-107.
[21] 李清泉, 郑年波, 徐敬海, 等. 一种基于道路网络层次拓扑结构的分层路径规划算法[J]. 中国图象图形学报, 2007, 12(7):1280-1285.
[22] Cooke K L, Halsey E. The shortest route through a network with time-dependent internodal transit times[J]. Journal of Mathematical Analysis and Applications, 1966, 14(3): 493-498.
[23] 孙海鹏, 翟传润, 等. 基于实时交通信息的动态路径规划技术[J]. 微计算机信息, 2007(24):177-178.
[24] 樊月珍. 基于交通流的车辆动态路径诱导方法研究[D]. 中国农业大学, 2005.
[25] Zhuang L , Gong J , He Z ,et al. Framework of experienced route planning based on taxis' GPS data[C].// Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on. IEEE, 2012:1026-1031.
[26] Li M , Wang W , Zhang Y . Research on driver experience based route planning method[C].// IEEE Intelligent Vehicles Symposium. IEEE, 2010:78-82.
[27] 唐炉亮, 常晓猛, 李清泉. 出租车经验知识建模与路径规划算法[J]. 测绘学报, 2010, 39(04):404-409.
[28] 戚欣, 梁伟涛, 马勇. 基于出租车轨迹数据的最优路径规划方法[J]. 计算机应用, 2017, 37(07):2106-2113.
[29] 谢海莹. 基于典型经验路径库的路径规划算法[J]. 交通运输研究, 2016, 2(01):17-22.
[30] 蒋新华, 朱丹丹, 廖律超, 等. 基于浮动车数据的道路单向限行状态动态识别[J]. 计算机应用, 2013, 33(06):1759-1762.
[31] 谭祥爽, 王静, 宋现锋, 等. 基于浮动车数据的路口探测方法[J]. 地理与地理信息科学, 2015, 31(05):34-38.
[32] 郑文斌. 基于浮动车轨迹数据的车道数量信息获取关键技术研究[D]. 武汉大学, 2017.
[33] 伍育红. 聚类算法综述[J]. 计算机科学, 2015, 42(S1):491-499.
[34] 廖律超, 蒋新华, 邹复民, 等. 浮动车轨迹数据聚类的有向密度方法[J]. 地球信息科学学报, 2015, 017(010):1152-1161.
[35] Palma A T, Bogorny V, Kuijpers B, et al. A clustering based approach for discovering interesting places in trajectories[C]. Proceedings of the 2008 ACM symposiumon Applied computing. ACM, 2008:863-868.
[36] Kharrat A, Popa I S, Zeitouni K, et al. Clustering algorithm for network constraint trajectories[M]. In: Headway in Spatial Data Handling. Springer Berlin Heidelberg,

2008:631-647.

[37] Rocha J A M R, Oliveira G, Alvares L O, et al. DBSMoT: A direction-based spatio-temporal clustering method[C]. Intelligent systems (IS), 2010 5th IEEE international conference, 2010:114-119.

[38] 周星星，吉根林，张书亮. 时空轨迹相似性度量方法综述[J]. 地理信息世界，2018, 25(04):11-18.

[39] 刘坤，杨杰. 基于编辑距离的轨迹相似性度量[J]. 上海交通大学学报，2009, 43(11):1725-1729.

[40] 樊东卫，何勃亮，李长华，等. 球面距离计算方法及精度比较[J]. 天文研究与技术，2019, 16(01):69-76.

[41] 刘旻，李梅，徐晓宇，等. 一种基于 HMM 模型改进的地图匹配算法[J]. 北京大学学报(自然科学版), 2018, 54(06):1235-1241.

作者简历及攻读硕士学位期间取得的研究成果

一、作者简历

2009.09-2013.06 就读于河北联合大学矿业工程学院，地理信息系统专业，获学士学位

2017.09-2020.06 就读于北京交通大学计算机与信息技术学院，软件工程专业

二、参与科研项目

[1]中国航天三院科技项目：智慧停车数据库及监控调度系统开发项目。

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：赵凤萍 签字日期：2020 年 6 月 15 日

学位论文数据集

表 1.1： 数据集页

关键词*	密级*	中图分类号	UDC	论文资助
路径规划；浮动车经验；轨迹聚类；HMM 算法；地图匹配	公开			
学位授予单位名称*		学位授予单位代码*	学位类别*	学位级别*
北京交通大学		10004	工学	硕士
论文题名*		并列题名		论文语种*
基于浮动车轨迹数据的路径规划研究				中文
作者姓名*	赵风萍		学号*	17127149
培养单位名称*		培养单位代码*	培养单位地址	邮编
北京交通大学		10004	北京市海淀区西直门外上园村 3 号	100044
工程领域*		研究方向*	学制*	学位授予年*
软件工程		数据分析与挖掘	2	2020
论文提交日期*	2020 年 5 月 4 日			
导师姓名*	李浥东		职称*	教授
评阅人	答辩委员会主席*		答辩委员会成员	
	瞿有利		赵守国、张大林	
电子版论文提交格式 文本（√） 图像（ ） 视频（ ） 音频（ ） 多媒体（ ） 其他（ ） 推荐格式：application/msword；application/pdf				
电子版论文出版（发布）者		电子版论文出版（发布）地		权限声明
论文总页数*	59			
共 33 项，其中带*为必填数据，为 21 项。				