本科毕业设计（论文）

基于移动大数据的信访人员识别和定位系统

Identification and Location System Based on Mobile Big Data

学　　院：计算机与信息技术学院

专　　业：计算机科学与技术

学生姓名：刘云朋

学　　号：16281229

指导教师：张晋豫

北京交通大学

2020 年 5 月 01 日

学士论文版权使用授权书

本学士论文作者完全了解北京交通大学有关保留、使用学士论文的规定。特授权北京交通大学可以将学士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。

（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：　　指导教师签名：

签字日期：　　年　　月　　日　　签字日期：　　年　　月　　日

中文摘要

摘要：随着无线通信技术的快速发展，日常生活中每日都有数以亿计的海量移动数据产生，对象移动行为呈多样性的同时仍保有规律性，故而，本课题依据移动基站提供的外来人员基站切换大数据信息，挖掘其行动路线轨迹，并根据它识别其中的上访人，给信访人员提供实时的位置信息。同时为了解决大数据存储计算和轨迹相似度匹配的问题使用了高低两层模型，低层模型将基于 MDL（最小描述长度）原则的分段划分降维和应用于线段聚类的邻居生长器算法（DBSCAN）相结合进行轨迹预处理以降低系统运算时间，高层模型将基于轨迹特征提取的二次划分加权降维和基于信息熵、时间插值与时间约束的改进版 Hausdorff 距离相结合进行轨迹匹配以提高相似度识别性能，以此构建大数据处理环境，最终形成在地图上呈现上访人员运动轨迹及其相似轨迹，同时据此判断其身份信息的 web 软件。

关键词：移动轨迹；分段降维；线段聚类；相似度匹配；身份识别

ABSTRACT

ABSTRACT: With the rapid development of wireless communication technology, hundreds of millions of mobile data are generated every day in our daily life, and the mobile behavior of objects is diverse while still maintaining regularity. Therefore, this subject switches the big data information according to the base station provided by the mobile base station, excavates its path of action, identifies the petitioner according to it, and provides the petitioner with Real time location information. At the same time, in order to solve the problem of big data storage calculation and track similarity matching, the high-level and low-level models are used. The low-level model combines the segmentation and dimensionality reduction based on MDL (minimum description length) principle with the neighbor grower algorithm (DBSCAN) based on segment clustering to preprocess the track to reduce the system operation time. The high-level model combines the two-level planning based on track feature extraction The improved Hausdorff distance based on information entropy, time interpolation and time constraint is combined to match the trajectory to improve the performance of similarity recognition, so as to build a big data processing environment, and finally form a web software that presents the petitioner's trajectory and similar trajectory on the map, and judges the identity information accordingly.

KEYWORDS： Moving track; Segment dimension reduction; Segment clustering; Similarity matching; Identity recognition

目　　录

1 绪论

1.1 研究背景及意义

随着无线通信技术的飞速发展，移动通讯设备的覆盖率日益增长，而城市间架设的移动基站可接收移动设备的数据信号，从而转化为海量的移动轨迹数据，研究人员可通过对这些数据进行收集、存储、过滤、融合、分析，以从中提取挖掘个体和群体的运动规律及社会化特征，使其具备更加重要的研究价值和实用意义，例如，本文涉及的对象间轨迹降维、聚类和相似度匹配，可具体应用于城市安全监管方面，辅助安全部门排查预警，在没有实时监控的情况下，根据其移动轨迹行为判断对象身份，可以极大地提高城市的安全系数。除此之外，此研究还可判断出相似人群，挖掘预测人群的兴趣点，以便应用于旅游业、餐饮业等更广大的市场，由此带来的经济效益不可估量。

1.2 国内外研究现状

1.2.1 轨迹降维技术研究现状

19 世纪 70 年代，由于信息化步伐的加快，现代轨迹降维技术开始逐渐涌现出来并得到了广泛的应用，这项技术的核心观点是通过更加精简的轨迹点来表达原始的轨迹特征，并期望得到尽可能小的差异，目前主要分为离线降维和在线降维两种降维方式。

离线轨迹数据降维方式是指将完整的轨迹作为算法的处理对象，也就是从已有的完整轨迹数据中提取轨迹特征，目前流行的 Douglas-Peucker 算法、均匀采样法、主要成分分析法、傅里叶变换法以及分段聚合法等都是采用这种思想的降维算法来对轨迹数据进行处理。其中，Douglas-Peucker 算法的特点是尽可能地优先筛选出关键特征点，具体操作时体现在重复采集轨迹数据中距离首尾轨迹点连线距离最大的且大于阈值范围的垂直距离点，该算法在减少轨迹点数量的同时，也可保留关键轨迹点，但是此算法需要提前测试选择距离阈值，阈值的选择对于后续实验精度有较大影响。均匀采样算法通过设置固定的采样时间间隔或者空间间隔，筛选有必要保存的轨迹点，组成近似轨迹，此算法是其中最直接的轨迹离线降维方式，简单且快捷，实现效率较高，并且保留的点都出自原始轨迹数据，降维后大幅度的偏移不太可能发生，但是此算法并未将轨迹点特征信息的重要性进行加权，容易造成某些关键轨迹点信息的流失而出现较大误差。主成分分析法思想是机器学习领域和数据科学领域十分常见的降维方法，以分解为特征向量矩阵

的乘积求出奇异矩阵，并由此遗弃一定的数据维度来达到降维目的，对应到轨迹数据降维中则是抛弃次要特征点达到压缩降维的目的，所以该降维方法可以较准确地提取轨迹特征点，也具有较高的鲁棒性，但是由于矩阵分解过程中的计算开销较大，使得其不适用于庞大的轨迹数据量。傅里叶变换降维算法与主成分分析法的思想类似，都是将数据从高维空间转换到低维空间，不同的是此方法是利用的傅里叶变换公式来降低数据的频域，转换出少量低频系数以此来降低数据量，同时有效地表示原始轨迹时序数据，该算法执行效率较高，但是对于噪声轨迹点比较敏感。分段聚合降维算法是根据制定的一些的规则对轨迹进行空间或时间上的划分，然后将各分段区间内的轨迹点做位置均值，期间可以考虑根据轨迹点的重要性进行加权聚合，以此作为降维轨迹的轨迹点。

在线轨迹数据降维方式是在运行过程中取得新的轨迹点时需要判断是否需要将其保留，其中主要的算法有滑动窗口算法以及蓄水池方法等。滑动窗口算法是通过判断窗口内轨迹点到首位连线的垂直距离是否小于阈值来进行取舍，有利于保留关键轨迹点，但此方法只关注窗口内的轨迹点，并没有考虑到全局情况。蓄水池方法针对的是实时生成的新轨迹点，算法通过等概率随机的方式挑选轨迹点加入到近似轨迹中，算法实现与结构都比较简单，而且效率极高，并且降维后有固定长度，然而缺点是不可以有效地保留关键轨迹点。

所以以上这些算法都存在诸如关键轨迹点流失、计算开销大、需提前设定阈值和噪声轨迹点敏感等缺陷。

1.2.2 轨迹相似度匹配技术研究现状

目前，轨迹间相似度度量方法和轨迹间基于相似度聚类是轨迹相似度匹配技术的主要实现方法。

轨迹间相似度度量采用轨迹间的距离来作为衡量标准，根据距离的不同定义又拓展出不同的计算距离的算法，主要有编辑距离算法、欧氏距离算法、最长公共子序列算法、Hausdorff 距离算法、以及动态时间弯曲算法。编辑距离算法的主要思想是计算使得一个字符串转换到另一个字符串的最少的操作次数，其中的操作步骤包括通过替代、插入以及删除等，此算法不要求轨迹点数量相同，但时空复杂度过高。欧氏距离是计算两组轨迹数据间的欧氏距离，即所有维度距离的平方均根，此算法构成简洁，并且使用方便，但缺点是要求轨迹必须具有相等的轨迹点数目，而且对噪声轨迹点十分敏感。最长公共子序列方法就是很直白地求出两轨迹点间的最长公共子序列，所以可以规避噪声点，但是计算开销过大。Hausdorff 距离方法是对于两个真子集的数据点，尽可能表达最不相似程度，即使用最大距离-最小距离来计算两点集间的相对距离，此算法可建立在轨迹点数目不相等的情况下，而且较为精确，但是对于噪声轨迹点还是较为敏感。动态时间弯曲

方法是在欧氏距离方法的基础上进行了改进和优化，有效降低了时间维度对于轨迹相似度计算造成的影响，对齐具有相似特征的轨迹点，实现时间维度的局部缩放，但是此算法的缺点是噪声点会对结果产生较大影响，而且由于算法复杂度高，十分消耗系统资源。

轨迹间基于相似度的聚类方法是通过设定某些关键的规则与参数，聚类与目标轨迹相似程度较高的移动轨迹，目前广泛采用的算法主要有密度聚类方法、层次聚类方法以及划分聚类方法。密度聚类方法是查找每条轨迹关于距离阈值 ε 可达的所有轨迹，并将其加入邻域，将领域内轨迹数目大于样本集合数目阈值 Minpts 的作为核心轨迹，其他作为噪声轨迹，以此进行轨迹聚类，此方法的有利之处在于可以有效处理噪声轨迹，并且有能力在聚类过程中处理任意形状的类簇，但是对于阈值设定较为敏感，并且当轨迹间密度不均匀时，聚类结果会受到较大影响。层次聚类算法的思想是通过分裂和凝聚构建决策树，即树结构应用当中的自顶向下分裂和自底向上合并，此算法并不需要提前设定阈值，能够挖掘轨迹间的层次关系，但是算法复杂度较高，而且过程中不能修改或撤销。划分聚类方法是提前随机选取 k 个初始质心，其他轨迹根据欧氏距离迭代分类，此方法易于理解，时间复杂度较低，但是对于处置敏感，且对噪声敏感，只能解决凸形数据。

所以，这些算法依然存在噪声轨迹点敏感、特殊值域敏感或计算开销大等不足。

1.3 研究目标和实验设计

1.3.1 研究目标

本文为了满足上访人员追踪定位的业务需求，包括人员识别、定位和追踪，构建移动大数据处理环境以解决海量数据的存储和计算问题，并提出合理的轨迹数据降维方法，在减少数据存储空间和计算量的同时有效保留轨迹运动特征，保证后续实验的精度。进而，根据本文使用的聚类算法快速筛选出相似轨迹集合，并根据改进过后的轨迹相似度匹配算法判断运动对象身份。最后，开发出基于 BaiduMap 的人员追踪识别 web 软件。

1.3.2 实验设计

1) 移动轨迹数据的收集和处理

针对轨迹数据收集，采取 LTE 网络架构收集数据与 BaiduMap 可视化追踪，通过对于用户移动轨迹的定位，收集并解析重要的轨迹结构信息，包括水平位置（纬度）、垂直位置（经度）和即时时间等静态信息，以及转角、方向角、速度和加速度等隐含信息，同时判断关键点，即兴趣点、停留点、移动点和拐点。在预处理过程中，清洗重复轨迹数据、无效轨迹数据，同时处理缺失轨迹数据。

2) 低层轨迹数据聚类模型

为了能减少系统在 web 运行计算负载，使用了提前对轨迹进行聚类的模型，快速排除大量的不相关的轨迹。而对于聚类算法的选择，为了减少轨迹整体聚类带来的特征信息缺失的问题，采取了基于 MDL 原则的分段划分降维和应用于线段聚类的 DBSCAN 算法来间接聚类原始轨迹，很大程度上提高了聚类的准确性。

MDL(Minimum Description Length)原则，是指当对其进行编码压缩时尽可能选择所需要保存的数据长度最小的模型。信息压缩的应用中此方法被广泛采用，因其可以很好的平衡轨迹分段的简洁性和精确性，在应用于海量移动轨迹数据处理及挖掘时具有良好的性能和效果。

轨迹聚类方法在 MDL 原则降维得到的轨迹线段集合中采取基于密度的轨迹线段聚类，尽可能快速地过滤掉大多数相似度较低的轨迹，减小后续匹配规模，且此方法具有较高的鲁棒性。

3) 高层轨迹相似度匹配模型

从目标轨迹所属的类簇中匹配与其最相似的轨迹，需要用轨迹间相似度识别的距离函数，这里依然采用 Hausdorff 距离，传统的 Hausdorff 距离将轨迹当作轨迹点集合来进行计算，忽略了轨迹的时间特征，而改进的 Hausdorff 距离改进了这一缺点，使之能更好地应用于轨迹相似度识别。同时更在二次划分加权降维减少数据量的基础上，采用了根据信息熵提取轨迹段和时间插值的方法进行辅助识别，进一步提高精确度。

基于轨迹特征提取的二次划分加权降维的算法思想是根据特定的空间距离和时间距离阈值，对等时间间隔划分后得到的子轨迹段集进行二次划分，由此轨迹的子轨迹段又可以由若干子段组成，最后通过计算各子段内所有轨迹点携带的特征信息的重要性，进行加权聚合，而不是简单地进行均值计算，以此得到能代表该轨迹子段的轨迹点。在聚合过程中，需要根据某轨迹点是否为兴趣点、停歇的时长占子段总时长的比例以及该轨迹点所在的空间维度坐标出现次数占子段轨迹点数量的比例，来为每个轨迹点设定相应的权重，之后进行加权聚合并更新聚合点的位置、时间及转角信息。

对原始轨迹进行二次划分加权降维后，根据信息熵提取目标轨迹中含有特征信息较多的时间划分子段——信息熵可以表达集合中数据点的混乱程度，映射到轨迹中即为在该时间段内移动对象活动的剧烈程度和运动轨迹的杂乱程度——并为待匹配轨迹提取与目标轨迹所提取的相同的时间分段。最后为轨迹数据完成时间插值操作和基于时间约束的改进版 Hausdorff 相似度计算，并为轨迹间的相似度进行评分。

4) 构建信访人员识别和定位系统

将研究总结到的移动轨迹数据获取、存储、降维、聚类以及匹配算法应用于实际轨迹数据和实际应用场景，构建基于 BaiduMap 的可视化系统。首先由 MongoDB 分片形式，构建大数据存储机制，高效安全的存储和管理数据。然后建立可以自动化处理、清

洗、融合轨迹数据，并且能够运行本文轨迹降维、聚类和匹配算法的后台环境。最后建立应用于实际场景的前台可视化用户界面，展示对于上访人员的定位和识别功能。

2 研究与分析

在进行具体的轨迹数据计算之前，需要先了解轨迹的基本概念以及轨迹数据的存储形式，以便后续对轨迹进行社会化特征提取等一系列相关操作。

2.1 轨迹基本概念

2.1.1 轨迹组成

轨迹数据显示了设备持有者某一时间段内的运动情况，移动互联网络能够根据无线信号定位设备所在地理位置，从而对其时空信息进行采样记录，进而连接收集到的采样点以构成移动设备持有者的移动轨迹数据。由这些移动轨迹点依照时间顺序排列，来近似表示移动对象的当前移动行为，所以，移动轨迹数据属于时序数据。

定义 2.1 移动轨迹：假设移动轨迹 Trail 是上述所描述的具有时间和空间属性的序列数据，则可以将其进行数字化表达，为$Trail = \{p_1, p_2, \dots p_n\}, 1 \le n \le \infty$，点 p 的信息包括二维空间的位置坐标<*x*,*y*>，以及采样时间 t，且$\forall\ 1 \le i \le n, t_i \le t_i + 1$。所以可将其描述为：

$$Trail = (< x_1, y_1, t_1 >, < x_2, y_2, t_2 >, \dots, < x_n, y_n, t_n >), 1 \le n \le \infty$$

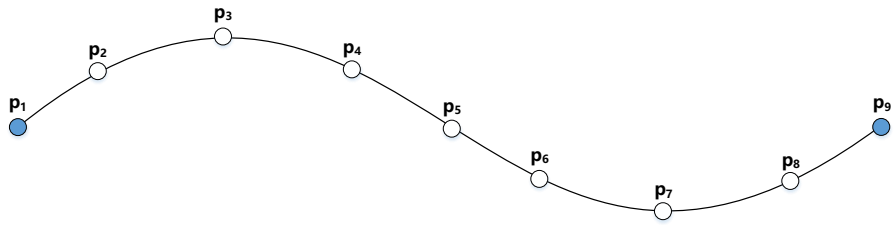


图 2-1 典型轨迹示意图

2.1.2 轨迹点结构

移动轨迹在直观上由点和线构成，用来描述对象的移动行为。同时，轨迹点的结构同样作为轨迹的重要组成部分，包含着大量的轨迹特征信息，分为静态信息和蕴含信息。静态信息是指可以直接从数据源直接提取出来的轨迹特征信息，如空间地理位置和轨迹点即时时间。蕴含信息是只需要在原始数据上进行计算分析才能得到的轨迹特征信息，例如轨迹形状特征和轨迹点速度特征，具体到算法需要和系统应用级别则包括轨迹点的停留时间、方向角、转角、速度和加速度等。下面是轨迹点信息结构的示意图：

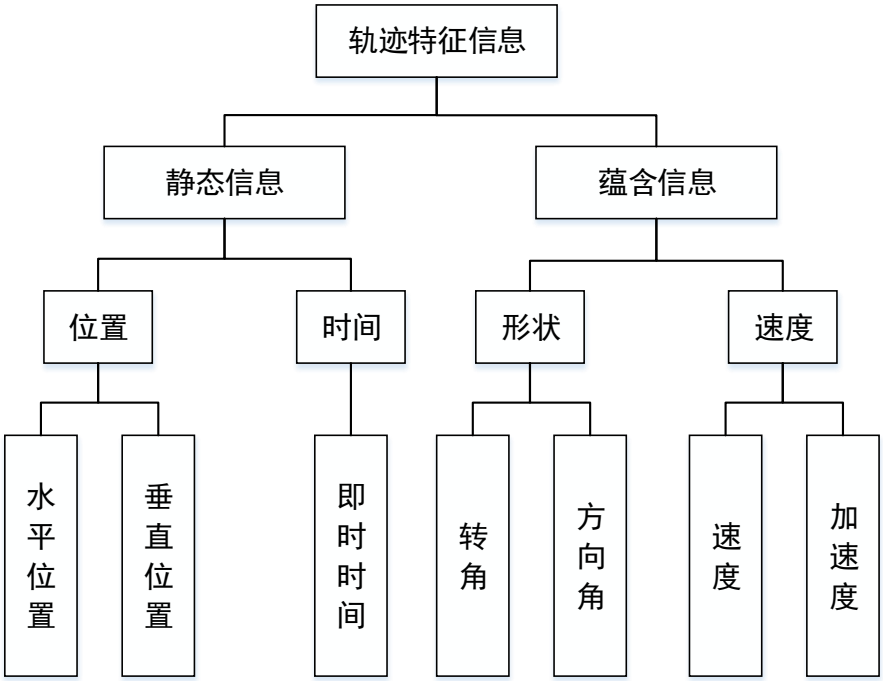


图 2-2 轨基点结构信息

2.1.3 关键轨迹点

移动轨迹有一系列具有时间和空间特征的轨迹点组成，但是并不是每一个轨迹点都在其轨迹中具有相同的重要程度，其中某个或某一组轨迹点有可能帮助研究人员决定最后结果。所以，在研究轨迹特征和轨迹点的过程中，可以使用算法分析和抓取具有关键信息的轨迹点，一方面可以应用于轨迹降维，降低数据存储量和计算量，另一方面可以提高计算结果的精确度。

停留点：用户在某一时段在某一地理位置保持静止状态，产生了长时间的停留，或者在某一区域速度明显减小，说明其对此区域产生了兴趣，可以说明此轨迹点具有比较明显的特征，或者可能成为用户身份识别的关键，所以在研究过程中需要进行提取或赋予较高权重。

兴趣点：用户在某一时间段内或者某一次运动时频繁经过某一地理位置，即轨迹点多次在某一小区域内聚集，可以说明此处的轨迹点集具有较高轨迹特征。

拐点：用户的运动过程本事复杂的折线表现形式，如果过程中突然产生较大的速度方向的改变，或者转角的突然加大，说明此处的运动具有明显的异常，蕴含较大的轨迹特征信息，可以将此命名为“拐点”，用于后续的轨迹特征提取之中。

同时，关键轨迹点的提取过程中并须使得精确性和简洁性两种相互矛盾的特性保持平衡。其中，精确性是指降维过后所保留的轨迹特征点数目不能过少，否则不能准确表达运动信息和归纳轨迹特征。简洁性的意思是说要在归纳轨迹特征的过程中要尽可能地

筛选出更少的特征点，保证足够的降维力度。可以看出这两个特性是互相排斥的，因此算法需要能够很好地找到这个平衡点。

2.2 轨迹数据

2.2.1 轨迹数据来源

本文涉及的轨迹数据由 LTE 网络架构生成。LTE 网络架构是 E-UTRAN 省掉 RNC 网络节点后的单层结构，这种做法便于简练网络和缩小延时，实现低复杂度、低延时、以及低消耗的要求。RNC 的功能并非被舍弃，而是被演进型 NodeB(Evovled Node B, eNode B)和服务网关(Serving GateWay，S-GW)所继承。如图所示：

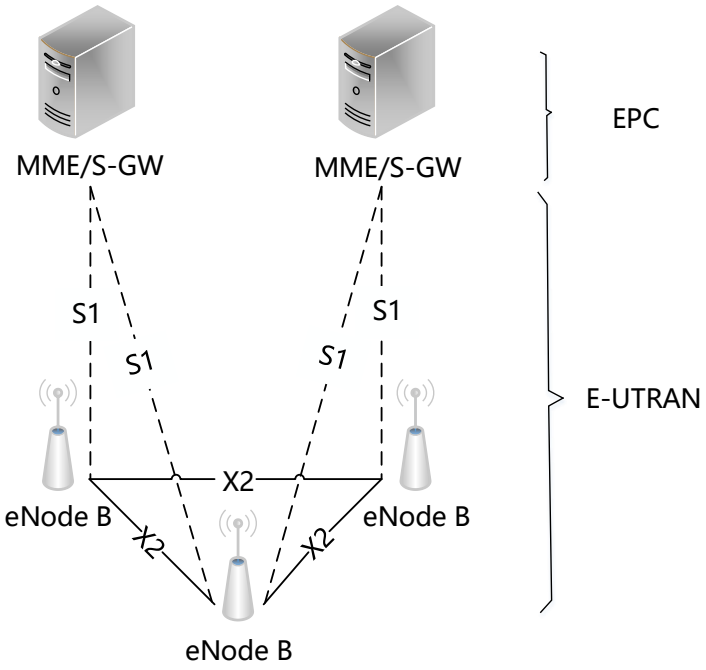


图 2-3 LTE 网络架构图

LTE 核心部件功能划分如下：

1) eNode B 功能：

- 无线资源配置：负责动态调配上行和下行的资源以及无线承载控制、无线许可控制和连接移动性控制等功能；
- IP 头压缩和用户数据流加密；
- 当无法在提供给 UE 的消息之内提取 MME 路由信息时，则转向选择从 UE 依附的 MME 中获取；

- 用户面数据向 S-GW 的路由；
- 由 MME 发出的呼叫消息请求的调度与发送；
- 由 MME 或 O&M 发出的广播信息的调度与发送。

2) MME 功能：

- 将寻呼消息发送到相关的 eNode B；
- 空闲状态的移动性控制；
- SAE 承载控制；
- 非接入层信令的加密和完整性保护。

3) S-GW 功能

- 用户数据包在无线接入网的终结；
- 支持 UE 移动特性的用户平面数据交互。

本文涉及的轨迹数据来自所持移动设备的上网数据，有用户移动设备在发出上网请求时 MME 对设备进行的定位，数据包含有基站位置信息以及时间信息，重要信息如下表所示：

表 2-1 MME 数据部分重要属性解析

属性名	属性说明	属性内容详例
Id	移动数据编号	387a987r9d990308
IMSI	移动设备唯一标识符	32842237757****
StartTime	基站接收信号即时时间	2018-04-29 16:34:32
Longitude	基站经度	136.23892
Latitude	基站纬度	26.28944
MME	MME 地址信息	100.10.254.21
E-UTRAN	基站 IP 地址	100.10.253.34
Cell-id	基站编号	28390
Local-id	小区编号	1
……	……	……

2.2.2 轨迹数据存储

本系统涉及的真实数据量庞大，需要构建大数据处理环境，在后台自动化收集、清理、存储轨迹数据，免去繁杂的人工操作。

2.2.3 轨迹数据预处理

移动对象所产生的轨迹数据通常和一般数据存在较大差别，主要是移动轨迹数据在空间维度内和时间维度内都保有连续性和顺序性，但是一般数据于时空维度是离散的孤立的数据点。同时由 LTE 移动基站获取的轨迹数据又存在重复、缺失和无效的情况，为了提高轨迹数据的质量，保证更优化的实验结果，需要在使用收集到的数据之前对其进行预处理，也就是进行包括轨迹数据的清洗、轨迹数据的集成、轨迹数据在内的变换。

重复数据、缺失数据和无效数据会对算法和实验的结果产生较大的影响，甚至严重情况下会使得系统出现异常，所以这些都是移动轨迹数据清洗过程中所需要着重分析处理的数据内容。重复数据是由于移动基站在接收到移动设备的接入请求消息时会为了确保设备接入成功而保留下多个信号，所以在接收到的轨迹数据中有可能会出现经纬度和即时时间都相同的数据，需要进行去重。缺失数据是由于信号受到干扰或信号转换紊乱造成的，如果缺失的是轨迹关键数据，如即时时间、经纬度、IMSI 设备号等，采用回归和线性插值等方式进行填充，若次要信息缺失，则考虑略过此属性或直接删除。

轨迹数据的集成是指将收集到的数据进行格式化规整，存储到数据库之中，方便后续提取研究。

轨迹数据的变换是指将收集到的离散的轨迹数据对象化，使之具备时空连续和顺序属性，构成具有语义特性的轨迹序列。

3 低层模型：实现轨迹聚类

在系统对目标轨迹和待匹配轨迹进行相似度计算之前，通过轨迹聚类缩小匹配规模，可以很大程度上减小系统的反应时间，得到最佳的用户体验。本模块考虑到轨迹的时空属性，采取了基于 MDL 原则的分段降维算法和应用于线段聚类的 DBSCAN 算法。

3.1 模型整体思想

当前投入使用的轨迹聚类算法主要分为两类：一类是将整条轨迹作为一个整体，而不对其做分段处理，直接投入使用进行聚类，不难看出如此做法使得每条轨迹只能划分到一个簇；另一类方法则是将轨迹分段划分应用到轨迹聚类之中，即根据特定的规则将一条轨迹分为多段，然后再为划分之后的每一个轨迹子段进行聚类，如此每条轨迹有机会被划分到多个簇。

从准确度上来判断，基于分段的轨迹聚类要由于基于整体的轨迹聚类，本文也是采用前者来进行计算。例如下图所示：

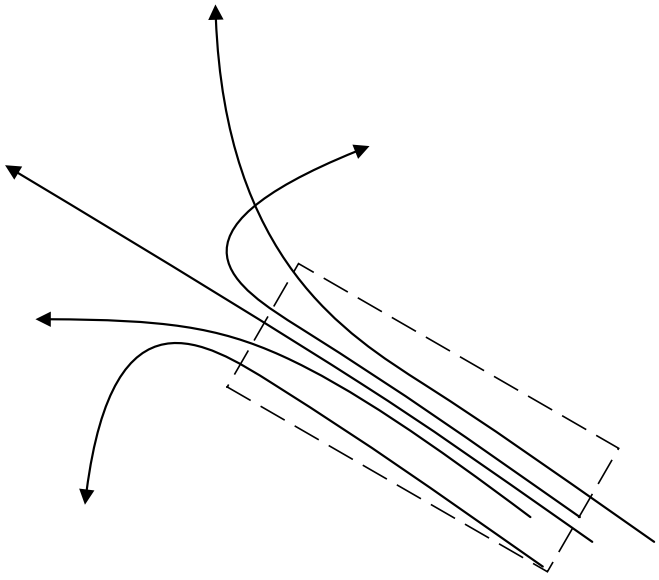


图 3-1 普通轨迹的例子

可以清楚地看到，如果我们按照轨迹整体进行聚类，由于五条轨迹都会去到不同的方向，那算法很难将他们聚类到一个类簇，从而忽略掉了矩形之内的很大一部分有价值的关键特征信息。

所以更好的方案是现将轨迹进行分段，在对于分段进行聚类，如下图所示：

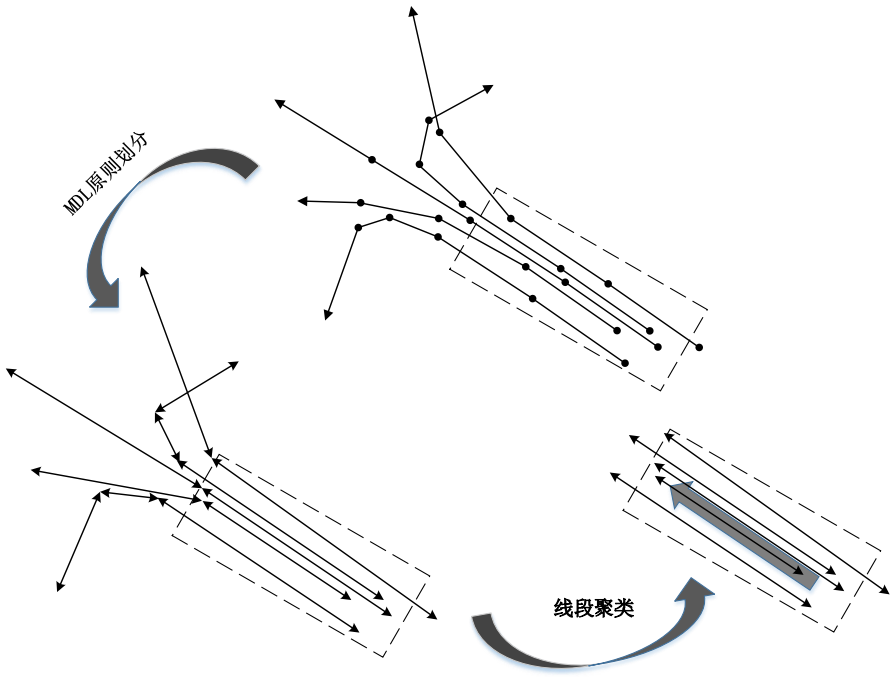


图 3-2 分段聚类的例子

首先将五条线段进行分段，转换成为多条线段的组合，再根据我们定义的距离度量函数将临近的线段进行聚类分组，然后根据线段所在类簇对原始轨迹进行标记，过程中可以将聚簇内的线段进行聚合生成一条代表轨迹，用于聚类方法的可视化和对于数据集外的轨迹进行匹配。

3.2 应用于线段聚类的轨迹距离方法

假设现在有两条线段，分别表示为$L_i = s_i e_i$和$L_j = s_j e_j$，并且L_i比L_j长，如下图所示：

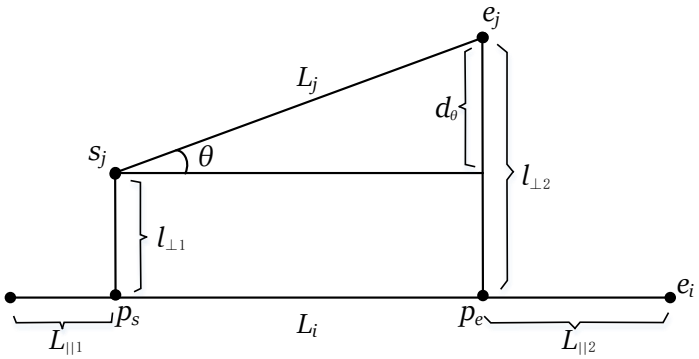


图 3-3 线段之间的距离

定义 3.2 线段间垂直距离：假设是p_s和p_e是线段L_j上的首尾端点s_j和e_j向线段L_i投影时与其的交点，$l_{\perp 1}$是点s_j到点p_s的距离，$l_{\perp 2}$是点e_j到点p_e的距离，即L_j两个端点到另一条线

段的垂直距离。计算公式如下：

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

定义 3.3 线段间平行距离：假设是p_s和p_e是线段L_j上的首尾端点s_j和e_j向线段L_i投影时与其的交点，$l_{||1}$是点s_i到点p_s的距离，$l_{||2}$是点e_i到点p_e的距离，即L_j的两个端点在L_i上对应的垂点与最临近的端点之间的距离。计算公式如下：

$$d_{||}(L_i, L_j) = MIN(l_{||1}, l_{||2}) \quad (3-2)$$

定义 3.4 线段间角距离：假设线段是有向的，$||L_j||$为线段L_j的长度，$\theta(0^o \le \theta \le 180^o)$是线段$L_i$和$L_j$的夹角，当其大于$90^o$时可看作两条线段具有很低的相似程度，假设不把轨迹的方向考虑在内，则线段间夹角相似度可以简单地使用 $||L_j|| \times sin(\theta)$ 进行度量。计算公式如下：

$$d_{\theta}(L_i, L_j) = \begin{cases} \left|\left|L_j\right|\right| \times \sin(\theta), & if\ 0^\circ \le \theta \le 90^\circ \\ \left|\left|L_j\right|\right|, & if\ 90^\circ \le \theta \le 180^\circ \end{cases} \quad (3-3)$$

三个定义中需要用到的交点p_s、p_e的空间坐标位置可以通过简单的几何计算公式得到，具体的公式如下：

$$p_s = s_i + u_1 \cdot \overrightarrow{s_i e_i},\quad p_e = s_i + u_2 \cdot \overrightarrow{s_i e_i}$$

$$where\ u_1 = \frac{\overrightarrow{s_i s_j} \cdot \overrightarrow{s_i e_i}}{\left|\left|\overrightarrow{s_i e_i}\right|\right|^2}, u_2 = \frac{\overrightarrow{s_i e_j} \cdot \overrightarrow{s_i e_i}}{\left|\left|\overrightarrow{s_i e_i}\right|\right|^2} \quad (3-4)$$

$$\cos(\theta) = \frac{\overrightarrow{s_i e_i} \cdot \overrightarrow{s_j e_j}}{\left|\left|\overrightarrow{s_i e_i}\right|\right| \cdot \left|\left|\overrightarrow{s_j e_j}\right|\right|} \quad (3-5)$$

此距离含义是在垂直方向、平行方向、旋转方向上的三种距离。对这三种距离线性叠加，得到的距离将会同时反映两条（线段）轨迹的位置、方向关系。整体距离定义及计算公式如下：

$$dist(L_i, L_j) = w_{\perp} \cdot d_{\perp}(L_i, L_j) + w_{||} \cdot d_{||}(L_i, L_j) + w_{\theta} \cdot d_{\theta}(L_i, L_j) \quad (3-6)$$

我们将$w_{\perp}$、$w_{||}$、w_{θ}三个权重的值初始化为 1。

3.3 基于 MDL 原则的分段划分降维

与 2.1.3 节中提到的关键轨迹点的提取问题相同，最佳的轨迹划分降维也同样需要考虑精确性和简洁性之间的平衡关系，因为如果所有的轨迹点都被选为特征点，那么精确度无疑就会达到最大值，但是简洁性会降到最低。同样，如果只有首轨迹点和尾轨迹

点被选为特征点，简洁性就会到达最大，但是精确度会降到最低。所以我们需要找到到两个性质的平衡点。

3.3.1 MDL 原则

最小描述长度(MDL) 原理是在对数据进行编码和解码的过程中，要求选择总描述长度最小的模型。

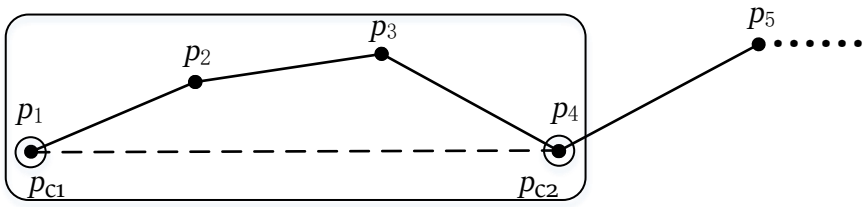
MDL 原则包含两个部分：

1. L(H): 描述压缩模型所需要的长度
2. L(D|H): 描述利用压缩模型所编码的数据所需要的长度

其中，H 代表估计，D 代表数据。解释数据 D 的最好的估计值 H 就是 L(H)和 L(D|H)加和的最小值。

应用到原始轨迹 Trail={p_1, p_2, p_3, p_4, …, p_n}和假设已经划分之后的轨迹{p_{c1}, p_{c2}, p_{c3}, p_{c4}, …, p_{cm}}中，则为如下定义：

如下图所示：



$$L(H) = log_2(d_\perp(p_1p_4, p_1p_2)+d_\perp(p_1p_4, p_2p_3)+d_\perp(p_1p_4, p_3p_4))+ log_2(d_\theta(p_1p_4, p_1p_2)+d_\theta(p_1p_4, p_2p_3)+d_\theta(p_1p_4, p_3p_4))$$

图 3-4 MDL 原则轨迹中的应用

此时，L(D|H)表示原始轨迹数据与划分轨迹数据差异性的总和。为了计算这个差异总和，我们需要求垂直距离和角距离的总和，而并不对其中的水平距离进行计算，这是因为闭合的两条线段间水平距离等于 0，所以并不做考虑。L(H)选择测量特征点之间的距离是为了排除在聚类过程中轨迹点坐标的影响。

最值得注意的是，这里我们用 L(H)代表上文提到的简洁性，而用 L(D|H)代表精确性。因为当分段数目增加时，L(H)会随之增大，而当原始轨迹和划分轨迹的差异值增大时，

L(D|H)会增大。所以划分过程需要做的就是利用贪心思想最小化(L(H)+L(D|H))的值。

3.3.2 轨迹分段降维

轨迹分段是在原始轨迹中选取一些特征点， 也就是地理位置、运动方向等特征改变剧烈的轨迹点。例如，对于轨迹 Trail={p_1, p_2, p_3, p_4, …, p_n}，找到一系列特征点{p_{c1}, p_{c2}, p_{c3}, p_{c4}, …, p_{cm}}，分成的轨迹线段即是连续两个特征点的连线，如下图所示：

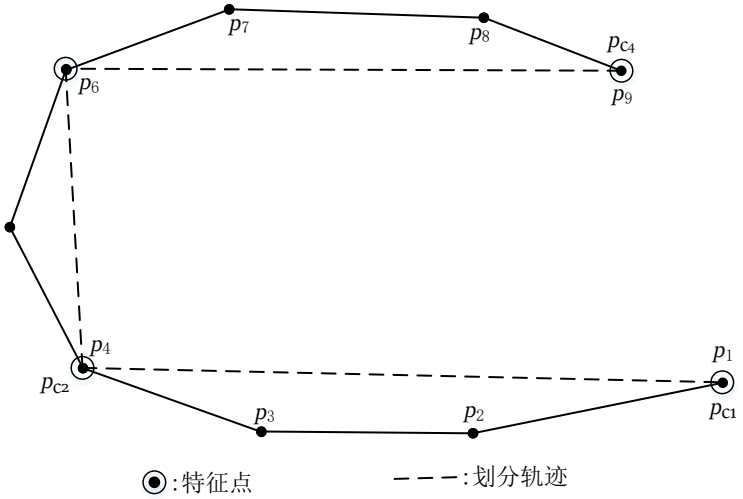


图 3-5 轨迹和轨迹划分实例

首先将首轨迹点作为特征点加入，向后遍历，判断是否选择将当前的轨迹点加入到轨迹特征点集合，如果加入后所产生的 MDL 开销值，也就是计算式$L(H) + L(D|H)$的值大于不将其加入的开销值，即$L(H)$的值，则将当前轨迹点的前一个轨迹点标记为特征点加入到集合之中，同时将当前轨迹点作为首轨迹点继续向后遍历，并重复以上操作，直到到达尾轨迹点，并将其加入特征点集合。

3.4 应用于线段聚类的邻居生长器算法（DBSCAN）

目前最经常使用的聚类算法共分为三类：一类叫做树形构造器，算法的思想是迭代合并距离最近的类，直到合成一个大类，这种方法也叫“阶层式聚类”(agglomerative hierarchical clustering)，它在最后阶段会构成的类呈树状。第二类是中心探索法，反复迭代并且重置中心点的坐标，直到构造出的所有的轨迹数据中心点与其所归属的类簇的距离总和不超过算法规定的距离阈值，这种算法又根据不同定义的中心点分为 K-Means，

K-Medians，K-Medoids 及模糊聚类(fuzzy clustering)。第三类即是本文采用的邻居生长器算法，将距离足够近的数据点连成一个类簇，直到所有数据点都被聚类，常用的算法有 DBSCAN，OPTICS 及 DENCLUE 算法。

邻居生长器是一种十分有效的聚类算法，它的特点是基于数据点的密度来进行聚类，特别的，本文采用的是邻居生长期算法中性能较好且符合本文轨迹聚类目标的 DBSCAN(Density-Based Spatial Clustering of Applications with Noise)。

3.4.1 名词定义

假设 D 是所有线段的集合，有如下定义：

定义 3.4 线段L_i到L_j的距离：根据上文介绍，再为每一项距离定义加上权重之后公式如下：

$$dist(L_i, L_j) = w_{\perp} \cdot d_{\perp}(L_i, L_j) + w_{||} \cdot d_{||}(L_i, L_j) + w_{\theta} \cdot d_{\theta}(L_i, L_j) \quad (3-7)$$

定义 3.5 线段的ε邻域$N_{\varepsilon}(L_i)$：已知$L_i \in D$及$L_j \in D$是数据集中的两条线段，线段L_j属于线段L_i的ε邻域当且仅当两线段之间的距离不大于ε，公式如下：

$$N_{\varepsilon}(L_i) = \{L_j \in D | dist(L_i, L_j) \leq \varepsilon\} \quad (3-8)$$

定义 3.6 MinLns 邻域最小轨迹数：已知$L_i \in D$近邻轨迹集中轨迹的数量表示为$|N_{\varepsilon}(L_i)|$，$MinLns$表示近邻轨迹集轨迹数量的阈值。

定义 3.7 核心线段：线段$L_i \in D$被标记为核心线段当且仅当其近邻轨迹集中轨迹的数量大于阈值，即$|N_{\varepsilon}(L_i)| \geq$ MinLns。否则被暂时标记为噪声线段。

定义 3.8 线段的直接密度可达：当且仅当线段$L_i \in D$属于线段$L_j \in D$邻域且L_j近邻轨迹集中轨迹数量大于阈值时，线段L_i直接密度可达线段L_j，即$L_i \in N_{\varepsilon}(L_i)$并且$|N_{\varepsilon}(L_i)| \geq$ MinLns。

定义 3.9 线段的密度可达：当且仅当存在一组线段$L_1, L_2, \ldots, L_n \in D$,并且$L_i$和$L_{i+1}$是关于$\varepsilon$和 $MinLns$直接密度可达时，则线段L_n是从L_1关于ε和 $MinLns$密度可达的。

定义 3.10 线段的密度连接：当且仅当存在一条线段$L_k \in D$使得线段$L_i \in D$与线段 $L_j \in D$均密度可达于线段L_k时，线段L_i密度相连于线段L_j

定义 3.11 线段的密度连接集：当且仅当一个集合D的非空子集C满足以下两个条件，它才可以作为密度连接集：

1）连接性：$\forall L_i, L_j \in C$，L_i密度相连于L_j

2）最大化：$\forall L_i, L_j \in D$，如果$L_i \in C$并且L_j密度可达于L_i，则有$L_j \in C$

如下图所示：

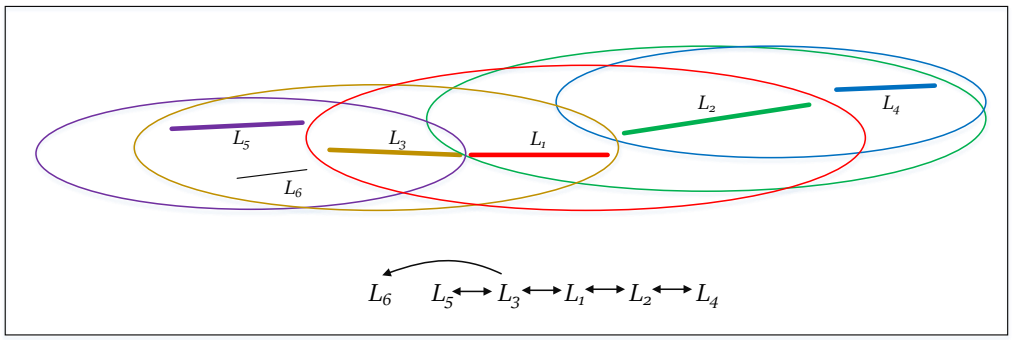


图 3-6 密度可达和密度连接

如上图，令$MinLns = 3$，线段的ε邻域用一个椭圆表示，可以推导出如下信息：

1）L_1, L_2, L_3, L_4, L_5均为核心线段；

2）L_2 (或L_3)直接密度可达于L_1；

3）L_6密度可达于L_1，然而L_1并不是密度可达于L_6的：所以说这种密度可达关系并非具有对称性，原因就是L_6并不是定义中的核心线段；

4）L_1, L_4与L_5均为密度相连：因为L_1, L_2, L_3, L_4, L_5形成了密度相连链。

3.4.2 算法过程

1）对于每一条未分类的线段，计算其邻域大小并判断是否为核心轨迹，若不是则继续搜索下一条线段，若是则执行以下步骤

2）将此核心线段的邻域线段加入队列，如果新加入的线段是核心线段且未被分类，则加入队列中进一步迭代扩展，如果不是核心线段则直接标记，不予以扩展。此步骤为迭代贴聚类标签的过程，可以同时更新原始轨迹的类标签。

3）计算每个簇的线段数目，若小于阈值则标记为噪声线段。

3.4.3 簇的代表轨迹生成

算法的主要思想是使用垂直于类簇中线段平均走向的直线逐次扫描各轨迹线段，每

当扫描到每条线段的首尾端点时都判断一下相交点的个数是否超过设定的 MinLns 阈值，如果超过阈值则进行相交点的平均计算得到聚合点，最终生成类簇代表轨迹。具体效果如下图所示：

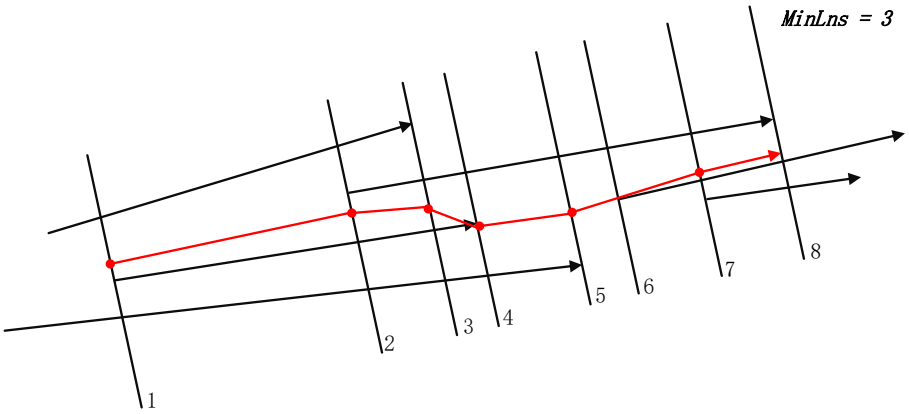


图 3-7 代表轨迹生成实例

3.5 实验

3.5.1 MDL 原则划分降维效果

测试降维力度的实验内容：从 18 万条轨迹中随机抽取 100 条同步进行均匀采样降维和基于 MDL 原则的划分降维，其中均匀采样降维是根据特定的时间间隔对轨迹点进行采样，生成降维后的轨迹。通过比较原始轨迹和两种降维方法的降维力度，即划分降维后的轨迹点的数目，分析基于 MDL 原则的划分降维的实际效果。结果如下图所示：

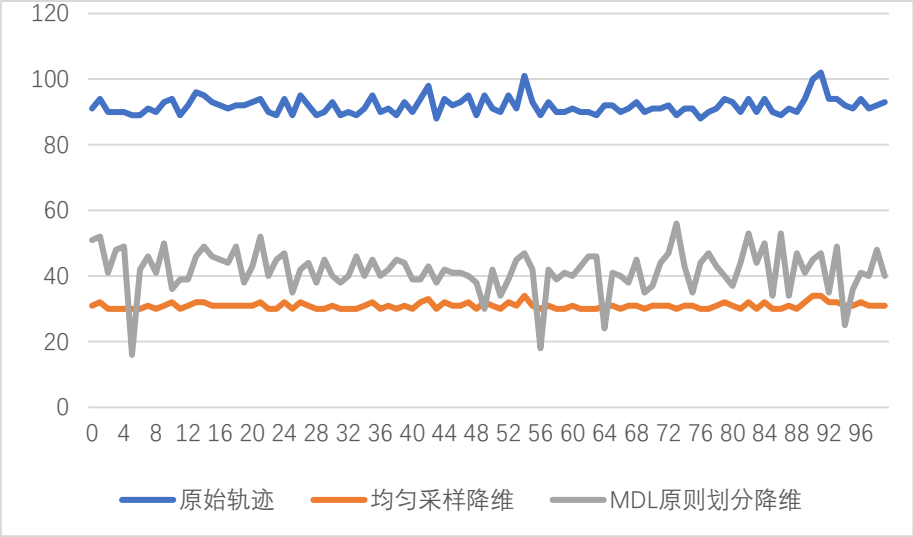


图 3-8 基于 MDL 原则降维力度

可见，基于 MDL 原则的轨迹划分降维具有很强的降维效果，接下来测试降维的特征保留效果，即通过降维之后与原始轨迹的 Hausdorff 距离之间的差距进行比较。

测试降维特征保留的实验内容：从 18 万条轨迹中随机抽取 101 条同步进行均匀采样降维和基于 MDL 原则的划分降维，计算其余 100 条与第 1 条原始轨迹的 Hausdorff 距离和两种方式降维后其余 100 条与第一条原始轨迹的 Hausdorff 距离，在统计与原始轨迹的差异，即

$$\frac{\left|Hausdorff_{原始轨迹} - Hausdorff_{降维轨迹}\right|}{Hausdorff_{原始轨迹}} \times 100\% \qquad (3-9)$$

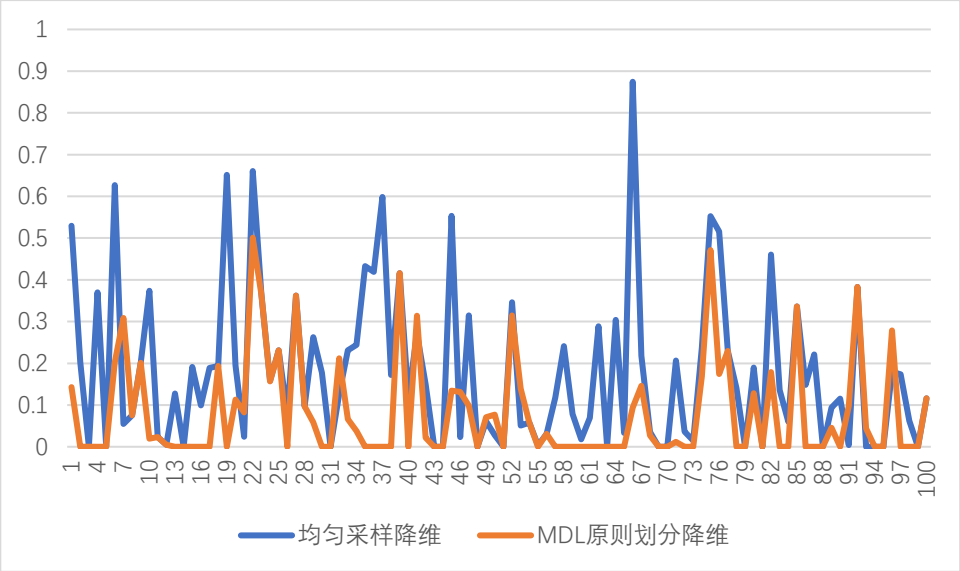


图 3-9 基于 MDL 原则降维特征保留

可见，基于 MDL 原则的划分降维的降维力度在与均匀采样降维相差不大的情况下，降维的效果明显优于后者，一是误差均值更低，二是误差峰值更低。

实际效果地图显示：

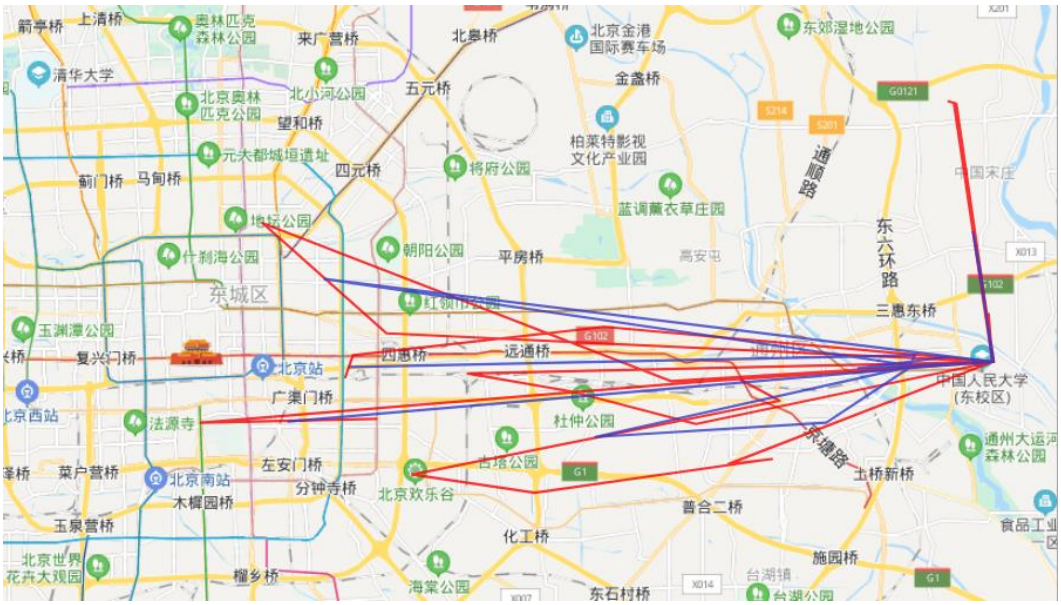


图 3-10 基于 MDL 原则降维效果地图展示

可见，基于 MDL 降维后的轨迹在地图上很好的描述了轨迹的运动特征，更具有明显的降维效果，同时优化了整体的存储空间和计算复杂度。

3.5.2 应用于线段聚类的 DBSCAN 算法

对于参数ε和$MinLns$的选择需要进行测试，由此找出最适合此实验数据集的数值。

表 3-1 $MinLns$的评测结果

参数	ε	$MinLns$	聚类数目	最小轨迹数	最大归属数	噪声轨迹数
	0.9	500	539	19	20	393
	0.9	1000	89	20	34	245
	0.9	1500	22	24	52	236

表 3-2 ε的评测结果

参数	ε	$MinLns$	聚类数目	最小轨迹数	最大归属数	噪声轨迹数
	0.88	1500	18	19	58	213
	0.90	1500	22	24	52	236
	0.92	1500	20	24	46	259

由上表可见，$MinLns$在等于 1500 时会有比较合理的聚类数目和可以接受的噪声轨迹，ε在 0.9 范围内的实验效果差距不大，但是我们选取 0.88 作为最后的结果，因为这时噪声轨迹数目较少。

表 3-3 轨迹整体和线段聚类比较

轨迹量级	时间	聚类数目	最小轨迹数	最大轨迹数	最大归属数	噪声轨迹数
20 万	1h35m	18	1	10,872	1	1,135
20 万	30m	20	19	12,556	58	213

由上表可见，线段聚类相比较以轨迹整体聚类来说，最明显的优势为运行时间短、同一条轨迹可以属于多个类簇、噪声轨迹少等优点。

4 高层模型：实现轨迹匹配

在进行轨迹聚类之后，可以快速的排除大量不相干轨迹，以便减少高层模型中轨迹匹配的运算量。在此模块中，首先对于轨迹进行了二次划分加权降维，再根据轨迹子段信息熵的大小继续提取关键轨迹，在减少数据量的同时提高关键信息段的比例。然后采取时间插值的方法提高轨迹间匹配的轨迹点数，以便更好地配合改进过后的 Hausdorff 距离计算方法。最终得到对象所在轨迹簇内各个相似轨迹的评分。

4.1 二次划分加权降维

现有轨迹 Trail={p_1, p_2, p_3, p_4, …, p_n}，根据时间划分为若干子轨迹段，转换为 Trail={$subTra_1$, $subTra_2$, $subTra_3$, …, $subTra_m$}，其中划分时间间隔 δ 相同，即每个子轨迹段时间长度相等。

二次划分是指根据设定时间阈值 λ 和距离阈值 l，将每个子轨迹段继续进行划分。鬼每个子轨迹段进行如下操作：首先将首轨迹点加入集合作为起始点，并以此向后遍历，如果当前轨迹点与起始点的距离大于距离阈值或者时间差值大于时间阈值，则将当前集合里的轨迹点进行加权聚合，并以此超出阈值的轨迹点作为新的起始点，继续向后遍历，并重复操作。最终对每个子轨迹段完成二次划分，$subTra_1$={p_1, p_2, …, p_{num}}

加权聚合是指根据需要聚合的轨迹点集内每个点的权重来进行聚合，这个权重是由停留点次数、停留时间和兴趣点的特征所决定的，即轨迹点连续停留在某一经纬度的次数占子轨迹段点总数的百分比 tp，轨迹点连续停留在某一经纬度持续的时间占子轨迹段总时间的百分比 lp 和某一经纬度在整个轨迹段内出现的次数占总轨迹点数的百分比 kp。权重公式如下所示

$$wp_i = lp_i \times tp_i \times kp_i \tag{4-1}$$

需要先将轨迹点集里的所有点的权重进行加和求出总权重，再将各轨迹点的经纬度乘以各自权重并求加和，由此计算出聚合点经纬度和时间，Points={p_1, p_2, …, p_{psum}}聚合轨迹点计算公式如下：

$$minp.lng = \frac{\sum_{i=1}^{psum} p_i.lng \times wp_i}{\sum_{i=1}^{psum} wp_i} \tag{4-2}$$

$$minp.lng = \frac{\sum_{i=1}^{psum} p_i.lat \times wp_i}{\sum_{i=1}^{psum} wp_i} \tag{4-3}$$

$$minp.t = \frac{p_1.t + p_{psum}.t}{2} \quad (4-4)$$

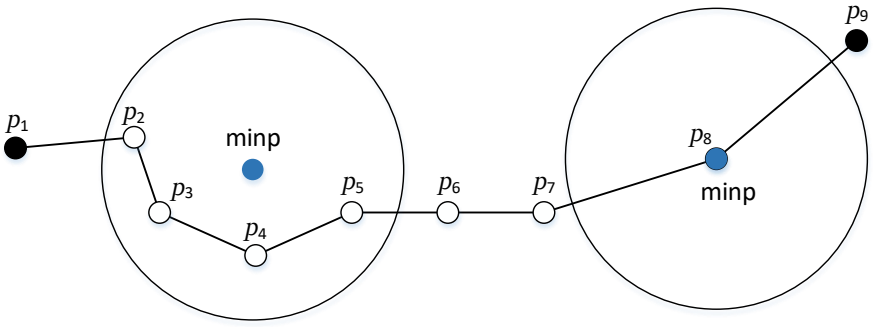


图 4-1 轨迹段二次划分

4.2 基于信息熵的移动轨迹段提取方法

4.2.1 信息熵度量

在信息论中，熵用来表示信息的不确定性的程度，不确定性越多，我们就需要越多的信息来了解，信息越多，不确定性也就越大。应用到移动轨迹之中，熵越大，表示轨迹运动越频繁，可挖掘的信息就更丰富。

目标轨迹按时间划分后表示为 Trail={$subTra_1, subTra_2, subTra_3, \ldots, subTra_m$}，以其中的 $subTra_1$={$p_1, p_2, \ldots, p_{num}$}为例，显示成经纬度坐标则有 Loc={$loc_1, loc_2, \ldots, loc_k$}，则它的信息定义为

$$l(loc_i) = -log_2 p(loc_i) \quad (4-5)$$

其中，$p(loc_i)$是 loc_i 在子轨迹段内出现的概率，在实验中以频率计算。

为了计算熵，需要计算所有经纬度包含的信息期望值，通过下面的公式得到：

$$H_j = \sum_{i=1}^{k} p(loc_i) log_2 p(loc_i) \quad (4-6)$$

其中 H_j 为该轨迹子段的信息熵。

4.2.2 移动轨迹段提取

通常情况下，并非所有子轨迹段都含有足够的轨迹特征，所以我们需要进行筛选，这样一来能够快速且高效的减少后续计算的运算量，同时能够提高移动轨迹特征信息提取的精确度。

将时间划分后的目标轨迹的所有的子轨迹段求出信息熵，选择所有熵值大于零的轨

迹段，并从大到小排序，并根据百分比β提取熵最大的子轨迹段，并保存其各个时间段，以便后续提取匹配轨迹的子段。

β的值需要实验调试，当β过大，保留的越多，达不到特征提取的标准；β过小，保留的越少，会损失过量的轨迹特征，严重影响轨迹匹配的精度。

4.3 轨迹时间插值方法

在某些情况下，不同轨迹同一时间段的轨迹点数量有较大差异，为了保证距离计算的精度，采取时间插值的方式填充轨迹点，即按照时间信息将目标轨迹和待匹配轨迹的基于信息熵提取的关键轨迹子段中的轨迹点进行一一匹配，插入的轨迹点坐标坐落在其前后两点的连线上。具体过程如下图所示：

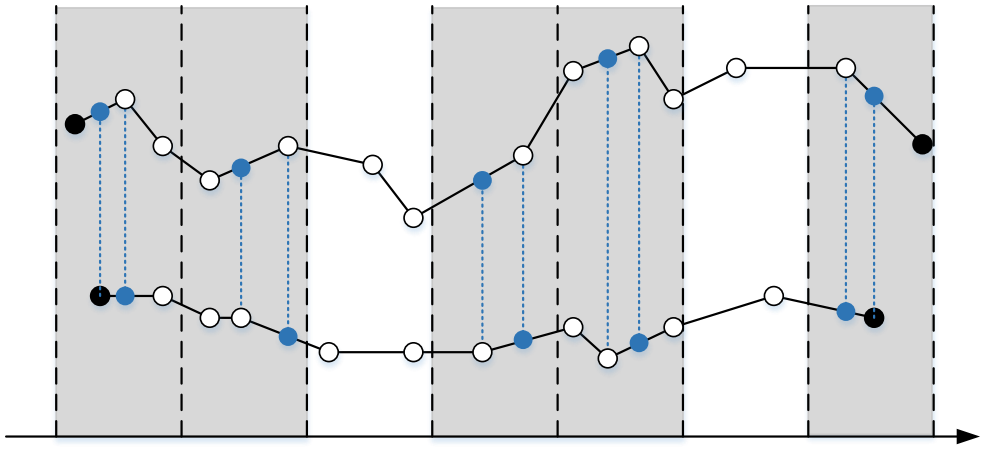


图 4-2 时间插值计算

4.4 基于时间约束的改进版 Hausdorff 距离

设X和Y是度量空间M的两个真子集，而$Hausdorff$距离$d_H(X,Y)$被定义为使得X的闭r-邻域覆盖Y，Y的闭r-邻域也覆盖X的最小的数r，具体数学公式表达如下：

$$d_H(X,Y) = \max\{sup_{x\in X} inf_{y\in Y} d(x,y), sup_{y\in Y} inf_{x\in X} d(x,y)\} \quad (4-7)$$

由以上定义的距离函数可得，度量空间就是M的所有真子集组成的集合，且记为$F(M)$。$F(M)$的拓扑也是依赖于M的拓扑。如果M是非空的，则$F(M)$也是。

对应到移动轨迹中：

$$H_k(subTraj_{ak}, subTraj_{bk}) =$$

$$\max\left(h_k(subTraj_{ak}, subTraj_{bk}), h_k(subTraj_{bk}, subTraj_{ak})\right) \quad (4-8)$$

$$h_k(subTraj_{ak}, subTraj_{bk}) = \max_{a_i \in subTraj_{ak}} \left(\min_{b_j \in subTraj_{bk}} dist(a_i - b_j) \right) \quad (4-9)$$

$$h_k(subTraj_{bk}, subTraj_{ak}) = \max_{a_i \in subTraj_{ak}} \left(\min_{b_j \in subTraj_{bk}} dist(a_i - b_j) \right) \quad (4-10)$$

$$H(Traj_a, Traj_b) = \frac{\sum_{k=1}^{Pset} H_k(subTraj_{ak}, subTraj_{bk})}{Pset} \quad (4-11)$$

但是，这种计算方式只是纯粹将轨迹按照轨迹点集合来进行距离计算，并没有考虑到轨迹的时间属性，并且当子段内轨迹点过多时，尤其是在时间插值之后，匹配范围过大更容易影响精确度。

所以解决办法是将每个轨迹点与待匹配轨迹的前一个和后一个轨迹点进行距离运算，改进后的定义如下：

$$H_k(subTraj_{ak}, subTraj_{bk}) =$$

$$\max\left(h_k(subTraj_{ak}, subTraj_{bk}), h_k(subTraj_{bk}, subTraj_{ak})\right) \quad (4-12)$$

$$h_k(subTraj_{ak}, subTraj_{bk}) = \max_{a_i \in subTraj_{ak}} (\min dist(a_i, (b_{i-1}, b_i, b_{i+1})) \quad (4-13)$$

$$h_k(subTraj_{bk}, subTraj_{ak}) = \max_{b_i \in subTraj_{bk}} (\min dist(b_i, (a_{i-1}, a_i, a_{i+1})) \quad (4-14)$$

$$H(Traj_a, Traj_b) = \frac{\sum_{k=1}^{Pset} H_k(subTraj_{ak}, subTraj_{bk})}{Pset} \quad (4-15)$$

由于之前采用了时间插值的方法，所以此根据轨迹的时间属性改进过后的 Hausdorff 距离并不会出现无法匹配$Traj_a$和$Traj_b$子轨迹段的轨迹点无法匹配的情况。

4.5 轨迹相似度评分

在将目标轨迹与所有待匹配轨迹进行基于时间插值的改进版 Hausdorff 距离计算之后，取出距离最小的 m 条待匹配轨迹，也就是相似度最高的待匹配轨迹，取出其中最大的距离H_{max}和最小的距离H_{min}，则第 i 条轨迹的评分如下：

$$Score_i = \left(1 - \frac{H_i - H_{min}}{H_{max} - H_{min}}\right) \times 100\% \quad (4-16)$$

4.6 实验

4.6.1 移动轨迹段提取

在对轨迹进行二次划分加权降维和根据信息熵提取关键轨迹段之前，需要对轨迹进行等时间段的划分，而时间间隔的长度需要实验来进一步分析确定。

实验内容：从 18 万条轨迹中随意选取 100 条轨迹，统计轨迹点数大于 5 的子轨迹段占总子轨迹段数的比例。

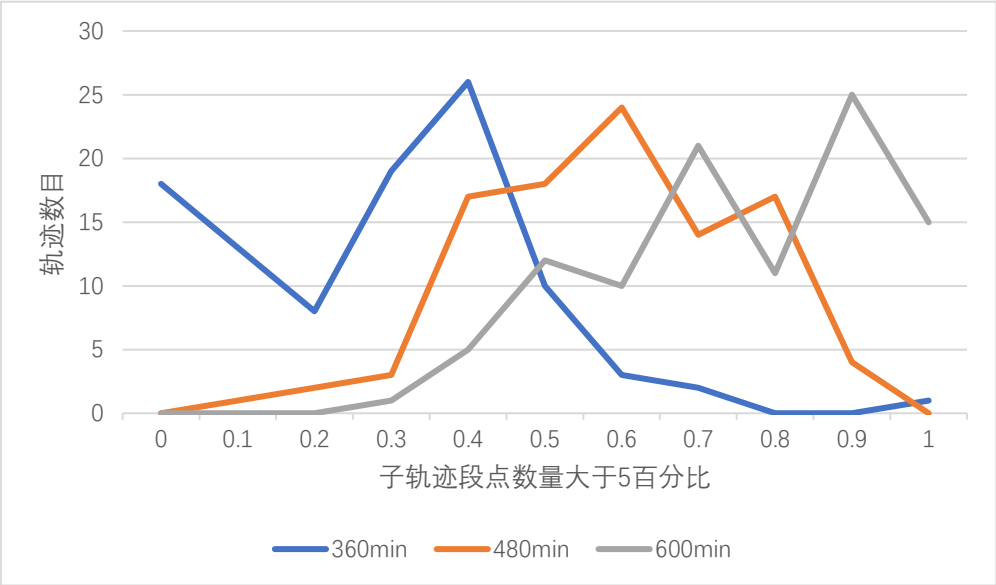


图 4-3 轨迹分时段划分时间间隔选取

可见，当时间间隔长度为 360 分钟时，大部分轨迹都只有 50%以下的子轨迹段含有 5 个及以上个数的轨迹点，也就是被过度划分了。当时间间隔长度为 600 分钟时，大部分轨迹都有 60%-100%的子轨迹段含有 5 个及以上个数的轨迹点，会存留较少的轨迹子段，如果进行二次划分降维时会损失大量轨迹特征，进行轨迹段提取时会存在轨迹段全提取的情况。而当时间间隔长度为 480 分钟时，大部分轨迹子轨迹段中轨迹点个数在 40%-80%之间，可以很好的平衡对于实验的需求。

4.6.2 二次划分降维效果

测试降维力度的实验内容：同 3.6.1 节实验内容

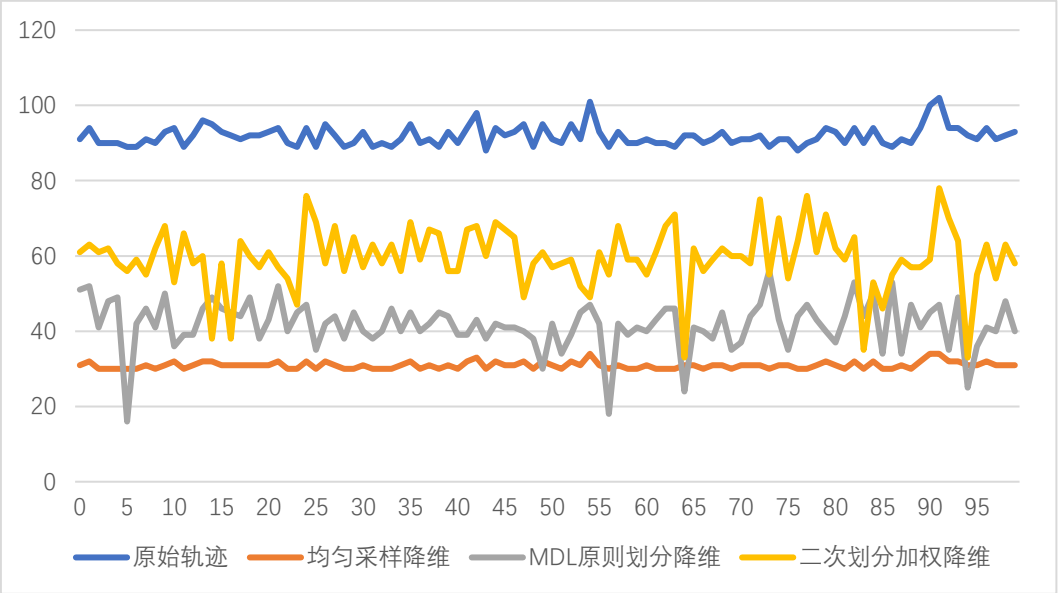


图 4-4 二次划分加权降维力度

可见，二次划分降维的降维力度比起 MDL 原则划分降维和均匀采样降维力度稍弱，但是用于相似度计算已经可以接受，因为相似度匹配时更注重轨迹特征的保留，所以需要进步一测试其轨迹特征保留程度。

测试降维特征保留的实验内容：同 3.6.1 节实验内容

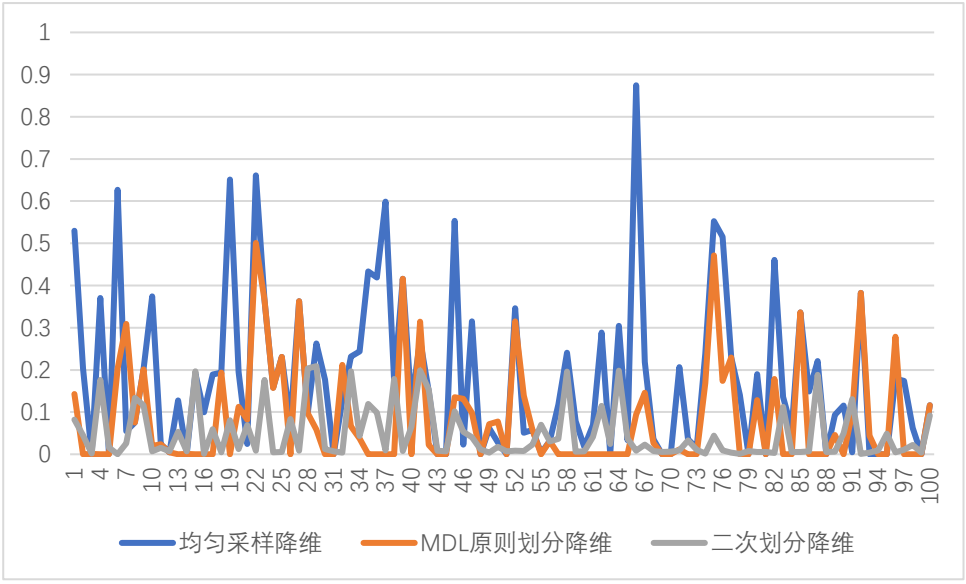


图 4-5 二次划分加权降维特征保留

可见，二次划分降维后轨迹之间的 Hausdorff 距离与原始轨迹之间对应的的 Hausdorff 的差距在 20%以下，可以满足相似度匹配算法对于轨迹特征保留程度的要求，并在一定程度上减少了计算量。

4.6.3 根据信息熵提取关键轨迹段的百分比

从 18 万条轨迹中随机选取一条目标轨迹，分比例根据信息熵提取关键轨迹段，另选 100 条轨迹根据目标轨迹的提取子段提取自身的关键轨迹子段，并求出与目标轨迹的 Hausdorff 距离的均值，通过与提取比例为 100%时对照，计算出不同提取比例下的计算精度，重复十次实验取平均值可以进一步保证实验的准确性以及有效性。

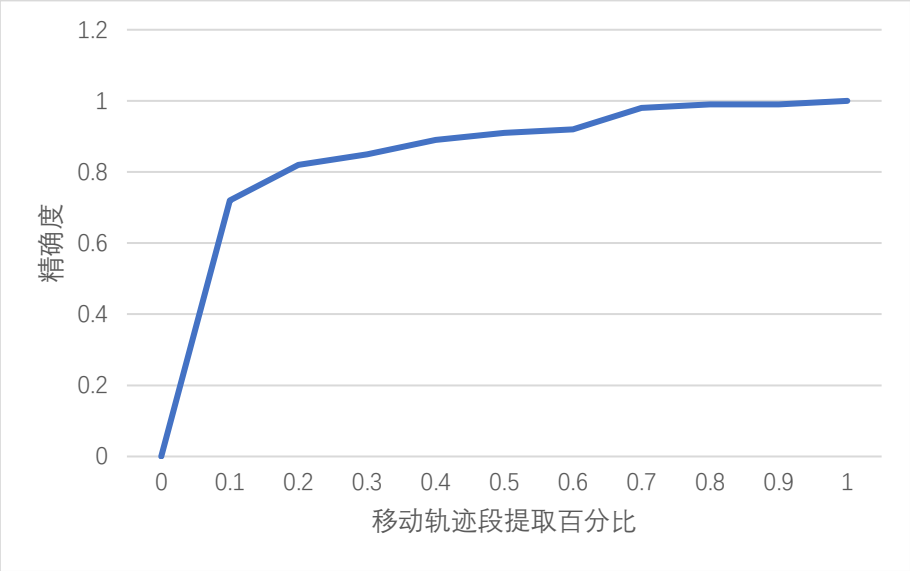


图 4-6 基于信息熵提取关键轨迹段比例

可见，当根据信息熵提取轨迹关键轨迹段时，提取比例在 70%以上已经可以达到 98%以上的精确度，所以在后续实验中可以选择 70%的提取比例，这种情况下可以在保证相似度匹配算法的准确性的同时尽可能的减少系统计算量。

4.6.4 改进版 Hausdorff 性能

算法的改进主要是针对时间性能，因为在一般情况下，传统的 Hausdorff 距离由于比较了轨迹中的所有轨迹点，所以会计算出比较小的轨迹间距离，但是带来的麻烦就是时间的消耗。考虑了轨迹点的时间属性，并不能明显的改进距离计算以及轨迹匹配的准确性，但是可以很大程度上减少计算量以及计算时间，直接从$o(n^2)$优化到了$o(n)$。

实验内容：从 18 万条轨迹中随机抽取 100 条，观察两种 Hausdorff 距离计算的结果及运行时间。

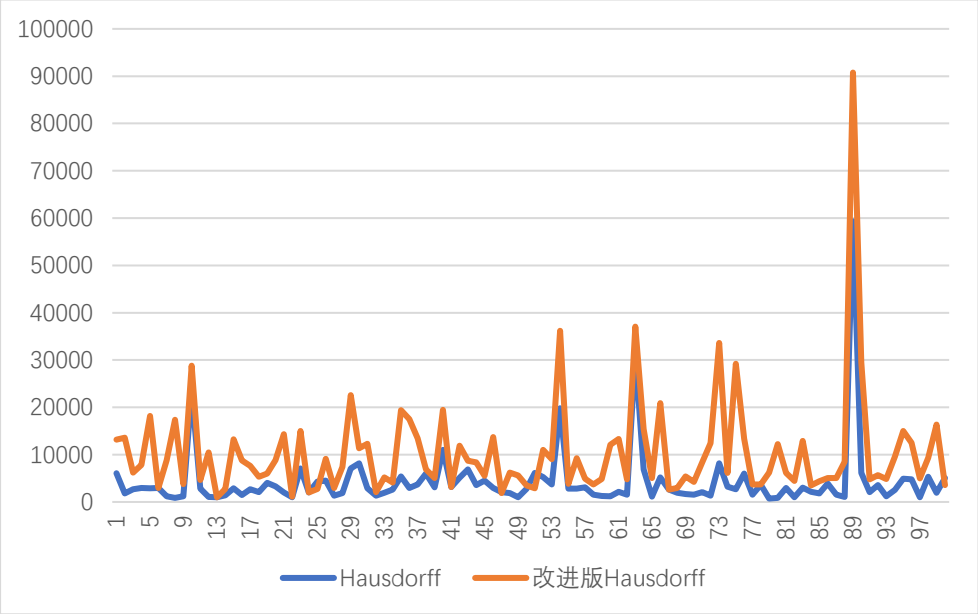


图 4-7 改进版 Hausdorff 距离与原始算法的结果比较

可见，改进之后的算法与原始算法之间是差距扩大的关系，具有相似的起伏效果，说明改进后的 Hausdorff 距离在一定程度上保留了原始算法结果的准确度，同时扩大了不同轨迹之间的差距，具有良好的性能。

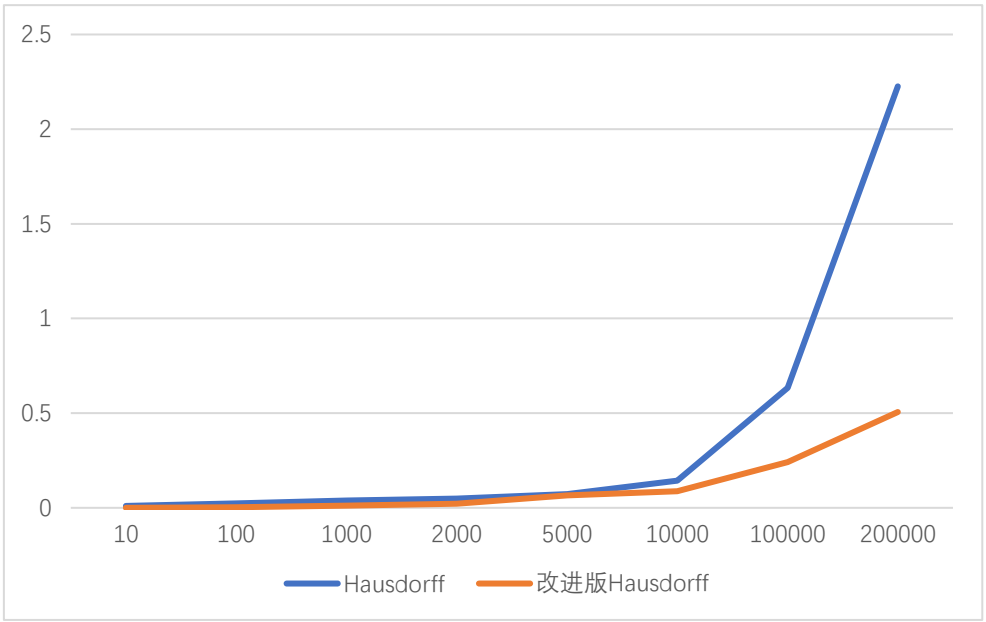


图 4-8 改进版 Hausdorff 距离与原始算法的运算时间比较

可见，当数据量逐渐增大，改进版算法具有更良好的性能，可以将计算量降低一个量级，当数据量增大时，效果更佳。

5 系统设计与实现

本系统的目的是根据移动基站所提供的轨迹信息，构建大数据存储机制，提取轨迹特征并识别上访人身份。所以系统需要解决的技术问题在于大数据存储、轨迹数据处理、相似度匹配算法，而重点与难点在于如何高效地对数据进行处理，降低系统运算时间的同时保障和提高系统的精确度，使其能够更准确地识别上访人。针对这些问题，本文构建了两层模型来对轨迹数据进行处理，低层模型对轨迹数据进行聚类，快速排除不相干轨迹，减少待匹配轨迹数据量，高层模型应用于实时查找匹配相似轨迹，并对 Hausdorff 距离进行了改进和优化以提高其在本系统中的适应性以及有效性。下面将具体介绍整个系统的运作流程。

5.1 系统简介

5.2 系统设计与实现

5.2.1 系统整体架构

整个系统由三部分组成：轨迹数据采集存储、低层轨迹数据聚类模型、高层轨迹相似度匹配模型。下图为流程示意图：

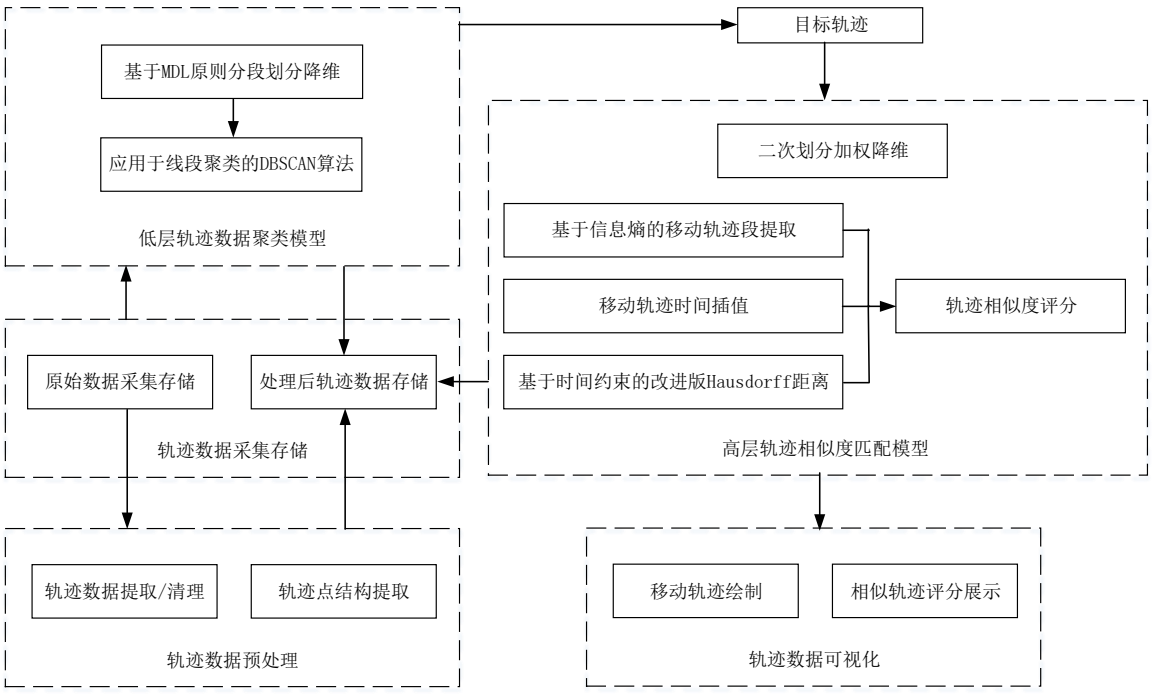


图 5-1 系统整体流程图

1）**轨迹数据采集存储：**轨迹数据来源于移动基站提供的 MME 移动设备轨迹数据信息，采集的数据会由于信号不稳定的因素出现缺失、重复和无效的情况，所以需要对数据进行清理。存储数据库使用的是 MongoDB，它是基于分布式文件系统建立的非关系型数据库，为 web 应用提供了高性能的数据存储能力，其分片功能同时为大数据存储和解决方案提供了极大的便利。

2）**低层轨迹数据聚类模型：**为了能减少系统在 web 运行计算负载，使用了提前对轨迹进行聚类的模型，快速排除大量的不相关的轨迹。而对于聚类算法的选择，为了减少轨迹整体聚类带来的特征信息缺失的问题，采取了划分和线段聚类的方法来间接聚类原始轨迹，很大程度上提高了聚类的准确性。

3）**高层轨迹相似度匹配模型：**从目标轨迹所属的类簇中匹配与其最相似的轨迹，需要用轨迹间相似度识别的距离函数，这里依然采用 Hausdorff 距离，传统的 Hausdorff 距离将轨迹当作轨迹点集合来进行计算，忽略了轨迹的时间特征，而改进的 Hausdorff 距离改进了这一缺点，使之能更好地应用于轨迹相似度识别。同时更在二次划分加权降维减少数据量的基础上，采用了根据信息熵提取轨迹段和时间插值的方法进行辅助识别，进一步提高精确度。

5.2.2 功能模块设计

系统的三个组成部分之中具体的功能如下图所示：

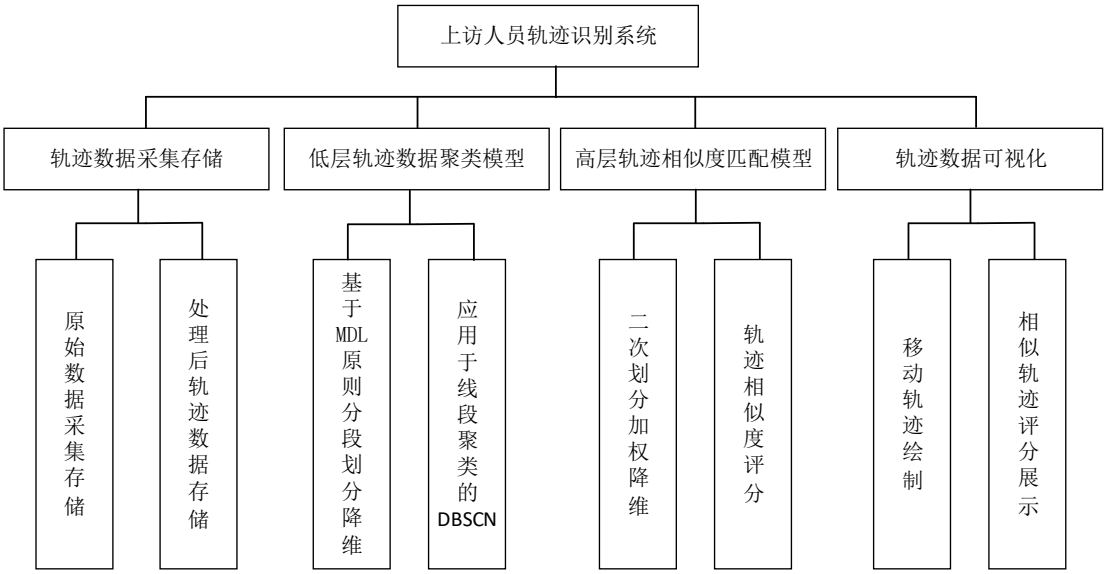


图 5-2 系统功能模块

5.2.3 数据库设计

为了构建大数据存储机制和提高系统的运行效率，需要实现 MongoDB 的分片功能。分片是 MongoDB 数据库的扩展功能，其根本动机是在不影响应用的性能的情况下，增加更多的机器或服务器来应对不停增长的数据和负载。其实质就是将数据进行分区，存储在不同的机器上，因此不需要功能强大的机器就能很好的处理更大的负载。

MongoDB 数据库目前已经实现了自动分片的功能，使人力得以从分片和切分数据的管理中解脱出来。具体的分片如下图所示：

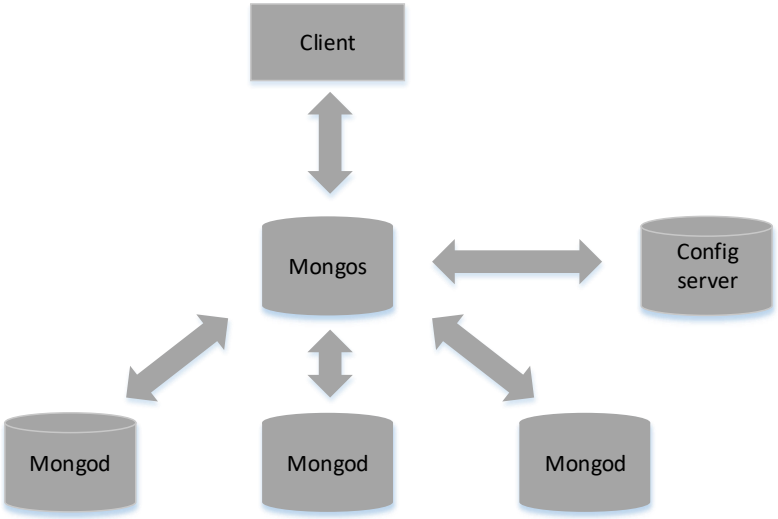


图 5-3 分片的客户端连接

1）Shard Server（Mongod）：用来存储来自实际应用中的数据。在现实的生产过程中可由几台机器组成一个 replica set 来担负 shard server 的功能，此种做法可以有效地防止主机单点故障。

2）Mongos：MongoDB 配置的路由器进程。它负责路由接受到的请求，随后聚合结果。而其本身并不存储配置信息和数据信息，但会缓存配置服务器的信息。

3）Config Server: 由于 mongos 并不永久存储配置数据，所以大部分集群的配置信息存储在 Config Server 中，也就是 Cluster MetaData，其中包括 Chunk 信息和数据与分片的对应关系。

4）Client: 客户端的接口，可以覆盖 MongoDB 自动分片的内在运行模式，从而使得前端应用能够被透明使用。

分片优势：

1）防止机器硬盘容量不足

2）可以将大量数据放在内存中提升系统性能

3）多个数据库服务器同时处理

数据存储格式：

表 5-1 移动轨迹数据存储格式

属性名	属性说明	属性内容详例
Id	移动轨迹唯一 ID	3ew2r43eu2398492739
Parent_id	原始轨迹唯一 ID	23480293847ed83833f
IMSI	移动设备唯一标识	123892480980**
Longitude	移动轨迹经度数组	[136.123,136.384,……]
Latitude	移动轨迹纬度数组	[26.123,26.384,……]
TraceTime	移动轨迹时间数组	[2018-02-03 12:10:10,]
Cluster_id	聚类标识数组	[2,3,……]

5.3 效果展示

本系统实现了在 BaiduMap 上显示指定人员的轨迹路线的功能，和显示所输入的指定人员的相似轨迹的功能，同时能够显示每条轨迹的相似度评分，信访人员可以根据给出的相似轨迹集判断上访人员身份。

5.3.1 轨迹绘制

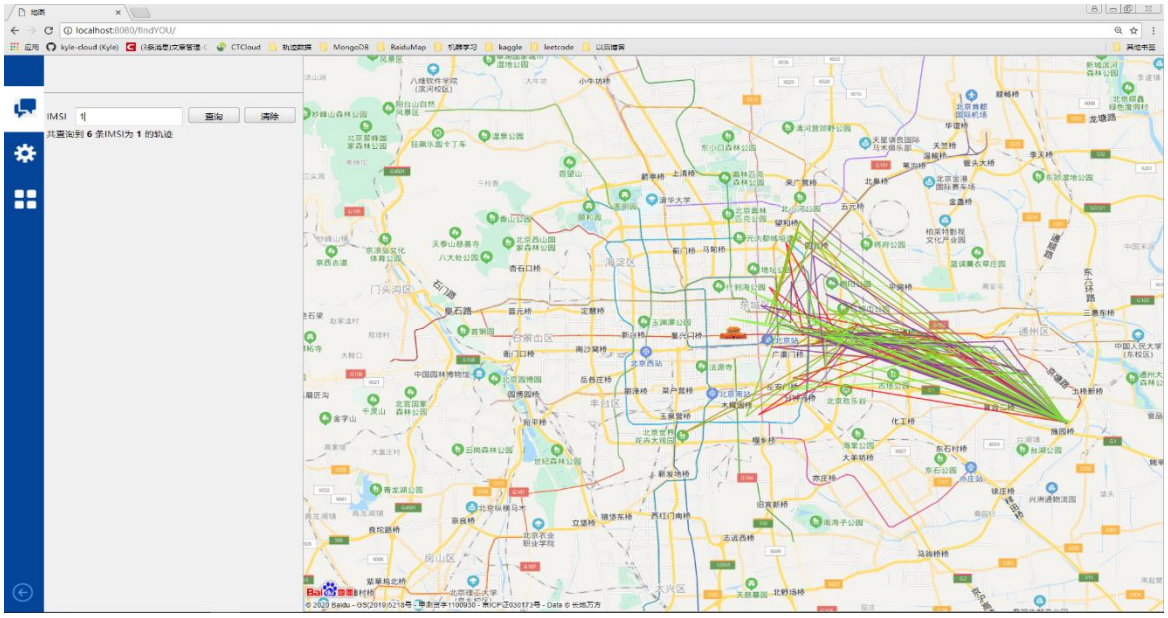
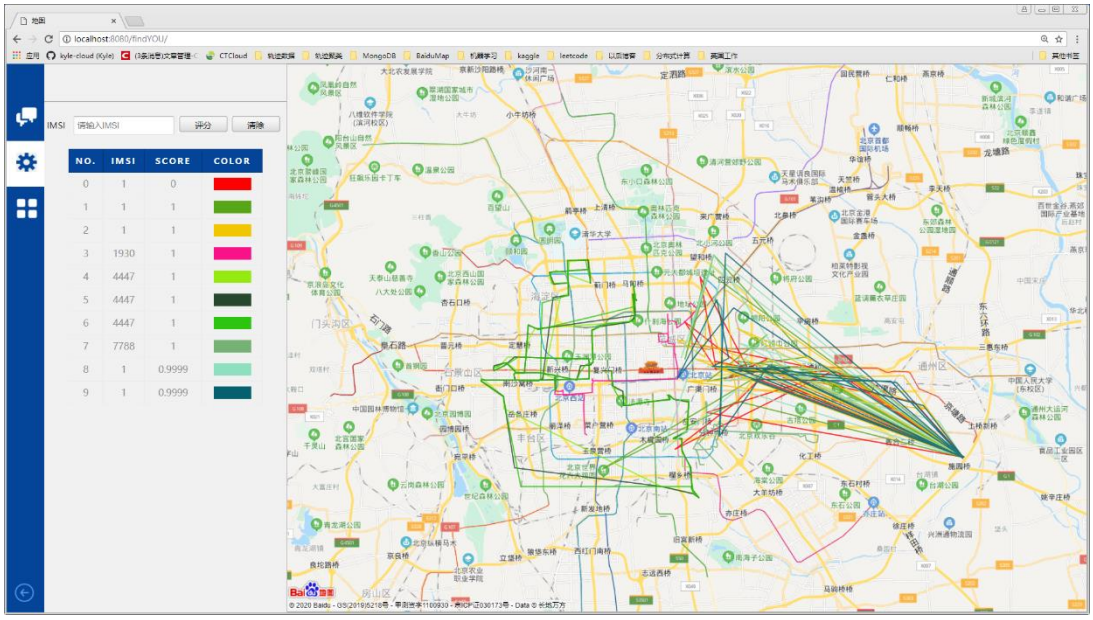


图 5-4 指定轨迹绘制

输入指定人员的 IMSI，在左侧显示运动轨迹的数目，地图上则展示移动轨迹的路线。图中红色为测试的轨迹。

5.3.2 轨迹相似度计算

输入指定人员的 IMSI，在左侧使用表格展示相似轨迹的对应信息与评分，地图上显



示相似的移动轨迹，红色为所输入的指定查找的人员轨迹。

图 5-5 相似轨迹绘制

可见，输入 IMSI 为‘1’的轨迹数据。在查找出的最相似的轨迹中，有四条轨迹 IMSI 为‘1’，三条为‘4447’，一条‘1930’，一条‘7788’，所以可以初步判断此用户有可能是其中的某个人，最有可能是‘1’，与我们所输入的用户相同。

参考文献

[1] 杨洁. 基于移动大数据的轨迹匹配算法研究[D].浙江工业大学,2017.

[2] 夏恬恬. 时空轨迹聚集模式挖掘算法的改进及并行化研究[D].武汉理工大学,2018.

[3] 杨东山. 多情景源异构环境用户运动轨迹分析[D].西安工程大学,2017.

[4] 曹妍妍,崔志明,吴健,孙涌. 一种改进 Hausdorff 距离和谱聚类的车辆轨迹模式学习方法[J]. 计算机应用与软件,2012,29(05):38-40+113.

[5] 邱运芬. 基于位置信息的移动对象行为模式分析[D].西南科技大学,2017.

[6] 郭鑫颖. 位置感知下的移动行为特征建模[D].西安科技大学,2017.

[7] 高瑞,周彩兰,朱荣. 移动轨迹挖掘算法设计与系统实现[J]. 现代电子技术,2017,40(01):134-136.

[8] 杨鹏程. 移动轨迹预测算法研究[D].北京邮电大学,2019.

[9] Jae-Gil Lee, Jiawei Han， Kyu-Young Whang. Trajectory Clustering: A Partition-and-Group Framework[Dissertation]

[10] González M C, Hidalgo C A, Albert-László B. Understanding individual human mobility patterns.[J]. Nature, 2008, 453(7196):779-82.

[11] Song C. Modelling the scaling properties of human mobility[J]. Nature Physics, 2010, 6(10):818-823.

[12] Lin M, Hsu W J. Mining GPS data for mobility patterns: A survey[J]. Pervasive & Mobile Computing, 2014, 12(11):1-16.

[13] Lv M, Chen L, Shen Y, et al. Measuring cell-id trajectory similarity for mobile phone route classification[J]. Knowledge-Based Systems, 2015, 89(C):181-191.

[14] 陈佳, 胡波, 左小清,等. 利用手机定位数据的用户特征挖掘[J]. 武汉大学学报信息科学版, 2014, 39(6):734-738.

[15] 江俊文，王晓玲. 轨迹数据压缩综述[J]. 华东师范大学学报自然科学版, 2015(5):61-76.

[16] 李海林, 郭崇慧. 基于云模型的时间序列分段聚合近似方法[J]. 控制与决策, 2011, 26(10): 1525-1529.

[17] 李正欣，张凤鸣，张晓丰等. 多元时间序列特征降维方法研究[J]. 小型微型计算机系统, 2013, 34(2):148-154.

[18] Nuha H H, Suwastika N A. Fractional fourier transform for decreasing seismic data lossy compression distortion[C]// International Conference on Information and Communication Technology. IEEE, 2015:590-593.

[19] Yu Y, Li J, Guan H, et al. Learning Hierarchical Features for Automated Extraction of Road Markings From 3-D Mobile LiDAR Point Clouds[J]. IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing, 2015, 8(2):709-726.

[20] Gómez-Conde I, Olivieri D N. A KPCA spatio-temporal differential geometric trajectory cloud

classifier for recognizing human actions in a CBVR system[J]. Expert Systems with Applications, 2015, 42(13):5472-5490.

[21] Gobbetti E, Guitián J A I, Marton F. COVRA: A compression-domain output-sensitive volume rendering architecture based on a sparse representation of voxel blocks[J]. Computer Graphics Forum, 2012, 31(3pt4):1315-1324.

[22] Zhang S. AIS Trajectories Simplification and Threshold Determination[J]. Journal of Navigation, 2016, 69(4):729-744.

[23] Keogh E, Chu S, Pazzani M. An Online Algorithm for Segmenting Time Series[C]// IEEE International Conference on Data Mining. IEEE, 2001:289-296.

[24] Li N, Sawan M. Neural signal compression using a minimum Euclidean or Manhattan distance cluster-based deterministic compressed sensing matrix[J]. Biomedical Signal Processing & Control, 2015, 19:44-55.

[25] Taha A A, Hanbury A. An Efficient Algorithm for Calculating the Exact Hausdorff Distance.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(11):2153.

[26] Xie X. A novel approach for detecting intersections from gps traces[C]// IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2016.

[27] Yuan Y, Raubal M. Measuring similarity of mobile phone user trajectories– a Spatio-temporal Edit Distance method[J]. International Journal of Geographical Information Science, 2014, 28(3):496-520.

[28] Jeong Y S, Jeong M K, Omitaomu O A. Weighted dynamic time warping for time series classification[J]. Pattern Recognition, 2011, 44(9):2231-2240.

[29] Zahra S, Ghazanfar M A, Khalid A, et al. Novel centroid selection approaches for KMeans-clustering based recommender systems[J]. Information Sciences, 2015, 320(C):156-189.

[30] Bouguettaya A, Yu Q, Liu X, et al. Efficient agglomerative hierarchical clustering[J]. Expert Systems with Applications, 2015, 42(5):2785-2797.

[31] Kumar K M, Reddy A R M. A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method[J]. Pattern Recognition, 2016, 58:39-48.

致　　谢

现在即将完成自己的毕业论文，心中有其实还是有那么一点点遗憾的，自己的项目其实还有很大的改进空间，还有很多优化方法都没有去实现，我还有很多想法。不过值得庆幸的是我并没有虎头蛇尾，而是在一开始就在认认真真的完成，从始至终都不曾有敷衍了事的心态，因为这是我真心想要去完成的，我也可以对自己说虽然自己展现出来的东西没有多么的高大上，但是按部就班的完成一件事情总是让人愉快的，就算是最后的一点小遗憾也是可以值得一辈子去怀念的。

在毕业设计的过程中，我很感谢张晋豫老师的信任，在我们交流的过程中，他总是在肯定我的想法，让我可以毫无顾忌的去展开，去实践，即使他没有当面教授我专业的技术，但是我在他的肯定中真的学到了四年来都想学却没有学到的新知识，也收获了更加广阔的视野。也正是他的肯定，允许了我像大海一样不断地去接纳、去吸收，让我知道了在我以后的道路上，有很多新奇的、美妙的知识与技术，现在，我可以说我已经改掉了畏难的性格，真的能够在兴趣的驱使下去克服一个又一个技术上的难关，这才是老师给我的最好的，毕业礼物。

刚刚读到我上铺何卫坑锋同学在他的致谢里说希望我早日完成大学四年来的目标改群名，想笑却又感动到泪目，所以我要给我亲爱的室友们写一些话。坑锋，你很优秀，事事都想着去争第一，也有能力去第一，关键还特别惜命，不管怎么说，希望你可以成为人生赢家，找到一份你喜欢又舒舒服服又能挣大钱的工作。下面就不分顺序了，谁让坑锋单独给我写了呢，我自然要把他单写。先说浩杰吧，莫名就有亲和力，不知道是不是因为他总是和女生在玩的关系，只玩不娶，但是一碰到他就很莫名其妙开启话痨模式，其实我不是的-_-!，希望浩杰可以一生平平安安，没有苦难，只有爱与被爱。子轩，嗯，太讨厌了，总是撅我，特别爱玩尤其是喜欢交很多很多的朋友，和我有些不一样，他还特别爱哭，大大咧咧的有特别敏感，嗯应该说是敢爱敢恨吧，喜欢就啵啵，不喜欢就骂街，希望他可以找到一份美好的爱情，当然我也是，然后和和美美的一生与他相伴。小治和江斧，嗯，按字典序的话，xiaozhi, jiangfu，先说小治，嗯没错。小治呢，我觉得是可以一起做很多事情的人，虽然我们俩没有特别的话题聊，但是总是想跟他一起出门，比如说吃饭啊、遛弯啊什么的，也不知道为什么，反正最想跟他一起出门我也不知道为什么，算了，反正就是希望小治可以和他的家人一生顺遂，永远幸福逍遥快活。恩最后到江斧了，应该是前两年我们那变化最大的了，因为某些原因开始颓颓佼，真是很想开骂啊，但是想到这个要收录，我就忍了，不过既然你想上北大，我知道确实挺难的，不过我还是支持你的，不管成不成功，至少你开始为了一个目标去改变了，我相信这是比

上北大更加有意义的，算了，挺你就是了，加油！

这一路，很感谢我的家人，朋友，老师们，他们是我奋斗拼搏的动力，也同样在支持着我前进，给我温暖、快乐。人生于世，总有那么一些人你愿意付出一切去守护，所有的温暖我都记在心中，永世不忘。

最后，谢谢你们，我毕业了。

附　录

附录 A 英文原文

Clustering of Vehicle Trajectories

Stefan Atev, Grant Miller, and Nikolaos Papanikolopoulos, *Fellow, IEEE*

***Abstract*—We present a method suitable for clustering of vehicle trajectories obtained by an automated vision system. We combine ideas from two spectral clustering methods and propose a trajectory similarity measure based on the Hausdorff distance, with modifications to improve its robustness and account for the fact that trajectories are ordered collections of points. We compare the proposed method with two well-known trajectory clustering methods on a few real-world data sets.**

***Index Terms*—Clustering of trajectories, time-series similarity measures, unsupervised learning.**

I. Introduction

THE trajectories of vehicles through a traffic scene are an information-rich feature that reveals the structure of the scene, provides clues to the events taking place, and allows inference about the interactions between traffic objects. Many vision algorithms are designed to collect trajectory data, either as their primary output, or as a by-product of their operation.

The focus of this work is to present a method for unsupervised clustering of vehicle trajectories with widely varying lengths. We augment the preliminary results presented in [1] with a comparison of the proposed method with two well-established approaches for trajectory clustering on several sets of trajectories for which manual ground-truth was obtained. The intended application for the method is as an extension to the traffic data collection system of [2], but automatic clustering has applications in collision prediction such as [3].

Researchers in the data mining and management fields are chiefly concerned with deriving similarity measures on trajectories. The sophistication of these measures is necessitated by the use of relatively simple clustering and indexing schemes, such as agglomerative clustering or k-means. Examples of this approach are [4], which uses the Longest Common Subsequence (LCSS) measure to cluster vehicle trajectories. The Edit Distance on Real Sequences presented in [5], [6] is closely related to LCSS. The need for compact representation and fast retrieval of trajectory data has resulted in dimensionality reduction approaches such as the segmented PCA in [7]. Another popular distance measure for trajectories and general time series is Dynamic Time Warping (DTW), exemplified by [8], where DTW is applied to a rotation-invariant representation of trajectories. In earlier work, [9] argued in favor of LCSS over DTW, a view shared by [10], where the same conclusion is reached. In both cases, LCSS only outperforms DTW significantly on data with large amounts of pointwise outliers that are ignored by the distance threshold of LCSS; the method proposed in this paper is similarly robust to the presence of outlying points, but that aspect turns out not to be important for our experiments, where DTW clearly outperforms LCSS.

Our work is similar to that of [11], who compare the unmodified Hausdorff distance, LCSS and DTW on a very similar task to the one considered here. In addition, they test a plain Euclidean distance and a Euclidean distance after linear dimensionality reduction with PCA. The experiments with the last two distances require resampling of trajectories, which loses information (in the particular case speed information and spatial fidelity). We specifically do not attempt to compare methods that require fixed-dimensional trajectory representation as they require a commitment to a specific trajectory model or features. Our only assumption is that the spatial aspects of the problem are more important that the dynamic, that is, a vehicle's path through the scene is more important than the speed/acceleration. This assumption is implicitly made in DTW and LCSS as well, and we will not consider the necessary modifications to remove it from the methods. Another work that compares an unmodified Hausdorff variant with LCSS is [12] who find the measures to perform similarly when outliers are accounted for.

The first objective of this study is to compare DTW, LCSS, and a modified Hausdorff distance when applied to the task of clustering trajectories of vehicles at traffic scenes. Our focus is not on indexing and retrieval, but rather on unsupervised learning. The second purpose of this work is to investigate the performance of spectral clustering methods for the problem domain. As our results show, there is no undue computational burden when working with moderately sized data sets. The clustering method we use is a modification of the method in [13], which uses a symmetrically normalized graph Laplacian as the affinity matrix. Rather than use a fixed global scale, as [13], we have adopted the scale selection strategy suggested by [14], where local scaling was introduced to the same spectral clustering variant. The use of spectral clustering is motivated by the demonstrated good performance of various methods, summarized in [15]. Instead of using a distortion metric to optimize a scale parameter, as [13] do, we use it to detect the number of clusters in the data automatically. Finally, we incorporate the directed Hausdorff distance directly in the local scaling method of [14] rather than first symmetrizing the distance in order to fit the framework of that work.

All of the approaches compared in this paper are similarity-based, use a non-parametric trajectory representation, and are invariant to the dynamic properties of the trajectories except

Manuscript received April XX 2008; revised ... This work was supported in part by the National Science Foundation through grants #IIS-0219863 and #IIP-0443945, the Minnesota Department of Transportation, and the ITS Institute at the University of Minnesota.

The authors are with the Department of Computer Science and Engineering, University of Minnesota–Twin Cities, Minneapolis, MN 55455 USA (e-mail: atev@cs.umn.edu, gmiller@cs.umn.edu, npapas@cs.umn.edu).

Digital Object Identifier 00.0000/TITS.2010.000000

for direction. Model-based (parametric) approaches allow for the normalization of trajectories of various lengths and number of points, but are not considered in this comparison as we want to evaluate model-free methods. Since we use the output of an automated tracking system operating at frame rates, we do not consider methods for resampling the trajectories. Different representations of trajectories are investigated in [16] where the Discrete Wavelet Transform, Principal Component Analysis, the Discrete Fourier Transform, and Linear Predictive Coding coefficients are all considered as normalization procedures for the clustering of time series. Hidden Markov Models have also been used to represent trajectories and time series, as in [17] and [18]. Finally, [19] exemplifies the modeling of univariate time series by auto-regressive models.

Applications that can benefit from unsupervised learning of trajectory patterns are demonstrated by [20], for use in an accident detection and prediction system. The extraction and clustering of trajectories is used for event detection by [21]. A relevant discussion of how trajectory information can improve tracking performance is provided in [22], where previously observed trajectories provide information to disambiguate subsequent tracking. In similar vein, [23] uses trajectory clustering to determine lane boundaries in a traffic video. The boundaries aid shadow detection and elimination, which ultimately improves subsequent tracking as well. In the work of [24], the trajectory information aids vehicle classification. A trajectory prior learned through clustering also enables anomaly detection, as exemplified in [25], who use the spectral clustering method of [13] without modifications.

The rest of this paper is organized as follows: In Section II, we summarize our data collection method, and include the definitions of LCSS and DTW for completeness. The proposed similarity measure for trajectories is developed in Section III, and the specifics of the clustering methods used are outlined in Section IV. The data sets and performance measure used are explained in Section V, while results are described in Section VI. We conclude with our remarks and proposed future work in Section VII.

II. Background

We start this section by briefly describing our data collection method. The tracking algorithm used is similar to the one used in our previous works [26], [27]. The remainder of the section will be used to summarize the LCSS and DTW similarity measures as used in the comparisons. A typical output of our tracking method is shown in Fig. 1.

A. Vehicle Tracking Method

We track vehicles in color and black-and-white video from static traffic cameras. The primitive features for the tracker are connected components extracted from the foreground, as identified by a background subtraction method that models each pixel as a Gaussian mixture with a few components. Before segmentation, illumination level filtering and video stabilization are applied to reduce artifacts and errors due to camera vibrations and sudden illumination changes. The

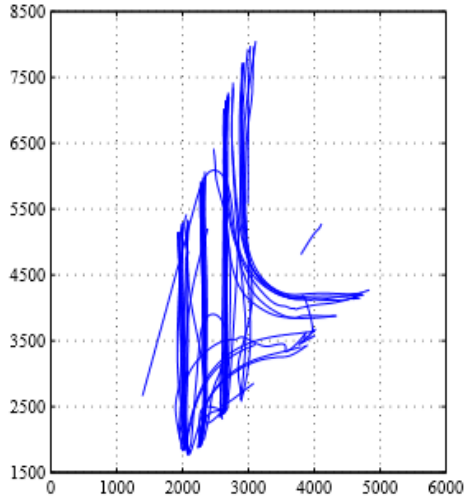


Fig. 1: A sample data set – trajectories collected at an urban intersection.

image-level tracking determines foreground region associations from one time instant to another based on fractional overlap, and uses some heuristics to detect region splits and merges that may indicate dynamic and static occlusions. Regions tracked without splits and merges for a few video frames are instantiated into targets for tracking, unless they already belong to a target.

The second level of tracking represents targets as sets of connected regions. Using camera calibration data, a 3D box model is fitted to the convex hull of each target's connected regions. The center of the box model projected on the ground plane is the vehicle location estimate, and the box dimensions are the vehicle size. While we do not use these measurements for classification purposes, we use the dimensions to filter out outliers (such as tracked pedestrians) and incorrectly initialized targets. The Extended Kalman Filter is used to accommodate the nonlinearity due to the perspective projection. In addition, once a target is determined to have left the scene, we use the intermediate output of the stochastic filter as input to the Kalman Smoother, thus ensuring that the location and size estimate at each time step is conditioned on all available measurements. As Fig. 1 shows, the trajectories recovered are extremely smooth and well-behaved. Finally, since the tracker operates at frame rates, the trajectories are sampled at a rate of 30Hz, which eliminates the need to interpolate them.

B. The LCSS and DTW Similarity Measures

The version of LCSS we use is defined in [4]. Given two scalar sequences $\mathbf{a} = \{a_i\}_{i=1}^{n}$, and $\mathbf{b} = \{b_j\}_{j=1}^{m}$, the one-dimensional LCSS similarity $\ell_{\epsilon,\delta}(\mathbf{a},\mathbf{b})$ is defined as follows:

$$\ell_{\epsilon,\delta}(\mathbf{a},\mathbf{b}) = \begin{cases} 0, & n = 0 \vee m = 0 \\ 1 + \ell_{\epsilon,\delta}(\mathbf{a}',\mathbf{b}'), & |a_n - b_m| < \epsilon \wedge |n - m| \leq \delta \\ \max\{\ell_{\epsilon,\delta}(\mathbf{a}',\mathbf{b}), \ell_{\epsilon,\delta}(\mathbf{a},\mathbf{b}')\}, & \text{otherwise,} \end{cases} \quad (1)$$

and 7^{th} nearest neighbors, so we kept a value consistent with our earlier work. In practice, the algorithm is much less sensitive to the choice of nearest neighbor than to the choice of global scale [14]. If the σ value is outside of the range 0.4 and 2 meters, we clip the value to the appropriate boundary. With that definition, we can write the final similarity measure we use as:

$$k(\mathbf{p},\mathbf{q}) = \exp\left(-\frac{h_{\alpha,w}(\mathbf{p},\mathbf{q})h_{\alpha,w}(\mathbf{q},\mathbf{p})}{2\sigma(\mathbf{p})\sigma(\mathbf{q})}\right), \qquad (20)$$

which is used whenever we need to use the modified Hausdorff distance in a spectral clustering framework.

At this point we would like to summarize the differences and similarities between the proposed distance and the LCSS and DTW methods. The actual distances computed by both DTW and the modified Hausdorff distance depend on an underlying metric directly, and have the same units as the point distance itself; in contrast, LCSS always returns a unitless distance that is based on a count. The one-dimensional LCSS distance between two series of length N can take on only $N+1$ distinct values as it ultimately depends on an integer count that can range between 0 and N. The length-normalization of the DTW distance (the factor of $1/n$ in (3)) is somewhat ambiguous when two sequences of widely differing lengths are compared as the normalization factor depends on the number of points of only one of the series, while the modified Hausdorff distance is based on a robust norm that needs no normalization by the number of points at all. The parameterization of LCSS used to control the "slack" in the matching is time-based (i.e., the parameter δ limits a correspondence neighborhood based on time), while the proposed Hausdorff distance controls the matching based on the arc-length, which is important when the series under comparison have longer periods without change (e.g., a vehicle stopped to wait for a traffic light does not change its position). While both DTW and the modified Hausdorff distance allow for a meaningful computation of relative distances, the LCSS distance is less discriminating because of the early application of the threshold ϵ. The time complexity of all distances in general is $O(mn)$.

IV. Trajectory Clustering

In this section, we describe the two clustering methods used, and also clarify the rules we use to decide which of the trajectories produced by the tracking method of Section II are considered outliers for our purposes. In the following exposition, d_{ij} will denote the distance between the i^{th} and j^{th} input trajectories (used by the agglomerative method), and k_{ij} will denote their corresponding similarity after appropriate scale selection (used in the spectral method).

A. Outlier Filtering

Since visual tracking can fail under various circumstances, not all automatically collected trajectories correspond to a single vehicle, and in extreme cases the objects being tracked are not vehicles at all. This is especially true for traffic intersections where tracking is very difficult in moderately dense traffic due to occlusions. Such outlier trajectories, if included in the input to the clustering algorithms, produce many singleton clusters and impede the learning process. We remove outliers in advance, to the extent possible. Since our tracking method produces dimension estimates and positions in real-world units, we have a sensible method for pre-filtering most trajectories generated by tracking failure or by merged targets.

A trajectory of a tracked object that violates any of the following rules is considered an outlier and not used in our experiments:

1) At least 90 samples (30 for highways) must be available.
2) The object length must exceed the width.
3) The object length must be between 1.5 and 6 meters.
4) The width must be between 1 and 4 meters.
5) The maximum instantaneous velocity over the trajectory must be between 10 and 100 kilometers per hour.
6) The objects' turning rate must be below $45°$ per frame.

A certain amount of trajectories that conform to all criteria listed above were considered outliers in the ground-truth. We kept these outliers in the input to the algorithms to test the methods' robustness. This is in contrast with [11], who eliminated all outliers manually in their experiments. Since none of the clustering methods detect outliers explicitly, we excluded the outlier trajectories from the clustering confusion matrix (that is, the score an algorithm received was not influenced by which cluster it assigned to the outliers).

B. Agglomerative Clustering

The agglomerative clustering method we use is identical to the one in [4] — each trajectory is initially assigned a singleton cluster. The two closest clusters are iteratively merged until all clusters are separated by a distance greater than a pre-specified threshold τ. If I and J denote the sets of indices of the trajectories of two disjoint clusters, the distance between the clusters is:

$$d_{\text{avg}}(I,J) = \frac{1}{|I|\,|J|}\sum_{i\in I}\sum_{j\in J} d_{ij}. \qquad (21)$$

In our tests, we specify the desired number of clusters rather than choosing a value of τ since the number of clusters is known from the manual ground-truth data. The range of values for τ that produce the same number of clusters is indicative of the sensitivity of the underlying distance measure and will be part of our experiments.

C. Spectral Clustering

As already explained, we use a variant of the clustering algorithm proposed by [13], but use the local scaling method of [14]. We include the description of the algorithm here for completeness. Note that we choose the number of clusters automatically using the distortion metric used in [13] to select the optimal global scale.

The input to the spectral clustering method is the affinity matrix $\mathbf{K}$ with elements k_{ij} for $1 \le i,j \le n$. The first step of the process is to normalize the rows and columns of $\mathbf{K}$, which yields a normalized affinity $\mathbf{L}$:

$$\mathbf{L} = \mathbf{W}^{-1/2}\mathbf{K}\mathbf{W}^{-1/2}, \qquad (22)$$

each panel. A histogram of trajectory lengths combined over all data sets is shown in Fig. 5e in order to illustrate the great variability in trajectory lengths (the x-axis is logarithmic).

C. Clustering Experiments

To the extent possible, we use equivalent parameters in all similarity measures. A particular problem with LCSS was the fact that for some choices of ϵ, the algorithm would be unable to produce the desired number of clusters since too many of the trajectories were at infinite distance from each other (the range for the LCSS distance in $\mathbb{R}^2$ is $[\sqrt{2}/2, \infty)$, but any two trajectories separated by a distance exceeding ϵ everywhere result in a distance of ∞). In such situations, we show the results of the algorithm using two different settings of ϵ – the first one is 12 feet (the lane-width in the scenes), and the second one is the smallest value of ϵ that was large enough so that the desired number of clusters was produced.

In cases where the automatically detected number of clusters differs from the true number of clusters, we present two scores for the proposed algorithm. To give an idea of the suitability of the proposed trajectory distance, we also used agglomerative clustering with the modified Hausdorff distance.

For LCSS and the modified Hausdorff distance, we used a slack parameter of one-half the trajectory lengths, that is, we set $w = 0.5$ for the modified distance and $\delta = 0.5$ for LCSS (very similar to the value in [4]). Finally, the parameter α for the Hausdorff method was set to 0.88, which was determined to be optimal on a subset of data set 1 in [1]. Note that the optimal α in [1] was chosen without recourse to the ground-truth data, it was the value that minimized the distortion score discussed in Section IV. The same value was used on all other data sets with good results and was not optimized for each of the remaining data sets.

VI. Results

The Variation of Information between the various clustering methods and the ground-truth is shown in Table II. The Clustering Error is also shown for clusterings that produce the same number of clusters as the ground truth. The actual number of clusters in the input is known to all algorithms, except for HS, DS, and LS-ϵ, which determine it automatically. The LCSS-based methods indicate the value of ϵ used as well, since it varied in one of the experiments. The best result for each distance measure (as measured by d_{VI}) is highlighted in *italics*, and the best overall result is highlighted in **bold**. The method names in Table II are:

Name	Distance	Type	Setting of k
HA	Hausdorff	Agglomerative	Manual
HS	Hausdorff	Spectral	Automatic
HM	Hausdorff	Spectral	Manual
DA	DTW	Agglomerative	Manual
DS	DTW	Spectral	Automatic
DM	DTW	Spectral	Manual
LA-ϵ	LCSS	Agglomerative	Manual
LS-ϵ	LCSS	Spectral	Automatic
LM-ϵ	LCSS	Spectral	Manual

A top-down view of the output of the proposed HS method is shown in Fig. 6.

A. Discussion

The first very clear result from Table II is that the modified Hausdorff distance better reflects the notion of trajectory similarity; even when used in an agglomerative fashion, the distance outperforms both DTW and LCSS. The modified Hausdorff distance also outperforms the other distances when used with the spectral clustering algorithm. The second observation is that when given the correct number of clusters, the spectral methods outperform the equivalent agglomerative method with the LCSS and Hausdorff distances. The fact that the LCSS distance used with larger thresholds does not benefit from the spectral clustering is a result of the unreasonably high thresholds used (which were necessary so that the agglomerative method produced the required number of clusters). The advantages of the spectral clustering are especially clear in data set 3, where the clusters contain many partially overlapping trajectories due to the widely varying starting point of successful tracking. While data set 3 appears very simple, the great variability in trajectory length and trajectory endpoints challenges all agglomerative clustering variants since the inter-cluster and intra-cluster distances both have large variances. The freeway data sets did not challenge the spectral method at all – both the number of clusters and class memberships were recovered perfectly. Surprisingly, HS outperformed HM on the second data set, even though it underestimated the true number of clusters (compare Fig. 5b and Fig. 6b). This can be explained by the fact that HS merged two clusters that had significant overlap, but otherwise produced results identical to the ground-truth for all other trajectories. Due to the overlap, HM also merged the same clusters, but it split another sparse cluster into two sub-clusters in order to obtain the specified 6 clusters.

The performance of LCSS for this problem, when used in the agglomerative fashion, is discouraging. When given a very reasonable ϵ threshold, equal to the actual inter-lane width, LA-12 failed to merge enough clusters to reach the specified correct number of clusters. The problem was that LCSS produced only two types of distances – very close to the minimum possible, and ∞, and any outliers or well-separated clusters produced many infinite distances. Increasing ϵ improved the performance of the method for the freeway data sets, but had the opposite effect on the intersection data sets, and affected negatively the spectral clustering results.

While the performance of DTW was not very good, we believe this is mostly due to the very strict constraints of (4) and (5) that require the starting points of the trajectories to be matched. Due to the unpredictability of target initialization in the tracker and the effects of foreshortening, trajectories in the same cluster could have varying starting and ending positions. Since the Hausdorff distance does not impose such a constraint, it has a clear advantage over DTW on data set 3, and to a lesser extent on data set 4. We were surprised by the poor results when using DTW in a spectral framework, and have no definitive explanation for this observation, except to note that on the data set with the largest intra-cluster variation in trajectory lengths, data set 1, DS and DM did not perform well, while the same methods outperformed DA for data set

TABLE III: Running time of the methods in seconds.

Data Set (Size)	Method	Similarity	Clustering	Total
1 (403)	HS	13.8	1.19	14.99
	HA	13.8	1.03	14.83
	LA-12	19.7	2.26	21.96
	DA	269.9	1.03	270.93
2 (391)	HS	1.44	0.62	2.06
	HA	1.44	0.91	2.35
	LA-12	1.56	2.08	3.64
	DA	20.6	0.92	21.52
3 (161)	HS	0.89	0.075	0.965
	HA	0.89	0.039	0.929
	LA-12	0.97	0.048	1.018
	DA	13.8	0.038	13.838
4 (240)	HS	0.81	0.18	0.99
	HA	0.81	0.10	0.91
	LA-12	0.88	0.27	1.15
	DA	11.6	0.10	11.7

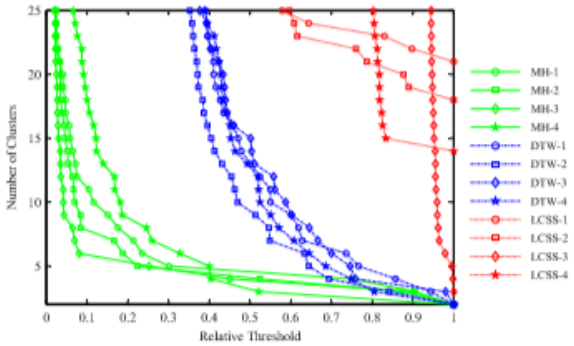


Fig. 7: A plot of the threshold in the agglomerative clustering versus the number of generated clusters. The modified Hausdorff (MH), DTW, and LCSS distance with $\epsilon = 12$ are shown for the four data sets. The threshold is relative to the largest finite value obtained (i.e., the ∞ of LCSS is ignored).

shows, LCSS is unable to merge clusters past a given point as all distances become infinite. Note that the steep portions of the plots for the modified Hausdorff distance occur prior to the correct number of clusters is reached for each of the four data sets. This means that the correct number of clusters may be easier to detect automatically using the Hausdorff distance by searching for the knee point on the cluster size vs. the threshold graph.

VII. Conclusions and Future Work

We presented a trajectory similarity measure based on the directed Hausdorff distance and applied it with both a spectral and an agglomerative clustering method. The imposition of some correspondence structure between the trajectories is a modification necessary to make the Hausdorff distance applicable for the comparison of ordered point sets, while the rejection of a certain amount of worst matches in the distance improves its robustness to noise and increases the tolerance to partial matches. Rather than use a distortion metric to select a globally optimal scale parameter, as [13] suggest, we used the metric to determine the optimal number of clusters from a set of possibilities obtained by examination of the spectrum of the affinity matrix. The local scale selection from [14] performed very well for the clustering of trajectories, hence there was no need to perform a search for any scale parameters. A possible future refinement of the spectral methods would be to allow for outlier detection.

It should be possible to combine the stricter correspondence of the DTW distance with the ideas of the proposed Hausdorff distance, as the strict monotonicity of the matching seems preferable for time series. In general, the average distance over corresponding pairs of points used by DTW seemed more likely to be influenced by outliers than the robust trimmed maximum we utilized. The relaxation of the starting/ending point constraints in DTW can also improve the suitability of the distance for matching trajectories with differing lengths and allow for some partial matches. However, it is not clear that both the monotonicity constraint of (6) and the trimmed maximum of (8) can be combined in a method of complexity $O(mn)$ for comparing an n-element and an m-element trajectory. We intend to investigate this question in the future.

The LCSS based distance measure performed disappointingly even on simpler data sets. Mostly, trajectories were considered very similar or very dissimilar, with no apparent middle case. This would make the LCSS distance unsuitable for use in spectral clustering with local scaling as the relative distances would be similarly clustered at a few extreme values. Using a robust loss in the DTW distance should allow for some of the desirable qualities of LCSS to be emulated.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable input and suggestions.

References

[1] S. Atev, O. Masoud, and N. P. Papanikolopoulos, "Learning traffic patterns at intersections by spectral clustering of motion trajectories," in *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, Oct. 2006, pp. 4851–4856.

[2] H. Veeraraghavan, O. Masoud, and N. P. Papanikolopoulos, "Computer vision algorithms for intersection monitoring," *IEEE Trans. Intelligent Transportation Systems*, vol. 4, pp. 78–89, Jun. 2003.

[3] S. Atev, H. Arumugam, O. Masoud, R. Janardan, and N. P. Papanikolopoulos, "A vision-based approach to collision prediction at traffic intersections," *IEEE Trans. Intelligent Transportation Systems*, vol. 6, no. 4, pp. 416–423, Dec. 2005.

[4] D. Buzan, S. Sclaroff, and G. Kollios, "Extraction and clustering of motion trajectories in video," in *Proc. Int'l Conf. Pattern Recognition*, vol. 2, Aug. 2004, pp. 521–524.

[5] L. Chen, M. T. Özsu, and V. Oria, "Symbolic representation and retrieval of moving object trajectories," in *ACM SIGMM Int'l. Wkshp. Multimedia Information Retrieval*, Oct. 2004, pp. 227–234.

[6] ——, "Robust and fast similarity search for moving object trajectories," in *Proc. ACM SIGMOD Int'l Conf. Management of Data*. New York, NY, USA: ACM Press, 2005, pp. 491–502.

[7] F. I. Bashir, A. A. Khokhar, and D. Schoenfeld, "Segmented trajectory based indexing and retrieval of video data," in *Proc. IEEE Int'l Conf. Image Processing*, vol. 2, Sep. 2003, pp. 623–626.

[8] M. Vlachos, D. Gunopulos, and G. Das, "Rotation invariant distance measures for trajectories." in *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, Aug. 2004, pp. 707–712.

[9] M. Vlachos, G. Kollios, and D. Gunopoulos, "Discovering similar multidimensional trajectories," in *Proc. 18th Int'l Conf. Data Engineering*, Feb. 2002, pp. 673–684.

[10] F. Duchene, C. Garbay, and V. Rialle, "Similarity measure for heterogeneous multivariate time-series," in *Proc. European Signal Processing Conference*, Sep. 2004, pp. 1605–1608.
[11] Z. Zhang, K. Huang, and T. Tan, "Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes," in *Proc. Int'l Conf. Pattern Recognition*, Aug. 2006, pp. 1135–1138.
[12] G. Antonini and J. P. Thiran, "Trajectories clustering in ICA space: an application to automatic counting of pedestrians in video sequences," in *Proc. Advanced Concepts for Intelligent Vision Systems*, Aug. 2004.
[13] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. Cambridge, MA: MIT Press, 2002, pp. 849–856.
[14] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems*, L. K. Saul, Y. Weiss, and L. Bottou, Eds., vol. 17. Cambridge, MA: MIT Press, 2005, pp. 1601–1608.
[15] D. Verma and M. Meilă, "A comparison of spectral clustering algorithms," Dept. Computer Sceience and Engineering, University of Washington, Seattle, WA, Tech. Rep. UW–CSE–03–05–01, 2003.
[16] K. Klpakis, D. Gada, and V. Puttagunta, "Distance measures for effective clustering of ARIMA time-series," in *Proc. IEEE Int'l Conf. Data Mining*, Nov. 2001, pp. 273–280.
[17] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, "Discovering clusters in motion time-series data," in *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, Jun. 2003, pp. 375–381.
[18] F. M. Porikli, "Learning object trajectory patterns by spectral clustering," in *Proc. IEEE Int'l Conf. Multimedia and Expo*, vol. 2, Jun. 2004, pp. 1171–1174.
[19] Y. Xiong and D.-Y. Yeung, "Mixtures of ARMA models for model-based time series clustering," in *Proc. IEEE Int'l Conf. Data Mining*, Dec. 2002, pp. 717–720.
[20] W. Hu, X. Xiao, D. Xie, T. Tan, and S. Maybank, "Traffic accident prediction using 3D model-based vehicle tracking," *IEEE Trans. Vehicular technology*, vol. 53, no. 3, pp. 677–694, May 2004.
[21] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, Aug. 2000.
[22] D. R. Magee, "Tracking multiple vehicles using foreground, background and motion models," *Image and Vision Computing*, vol. 22, no. 2, pp. 143–155, Feb. 2004.
[23] J. W. Hsieh, S. H. Yu, Y. S. Chen, and W. F. Hu, "Automatic traffic surveillance system for vehicle tracking and classification," *IEEE Trans. Intelligent Transportation Systems*, vol. 7, no. 2, Jun. 2006.
[24] B. Morris and M. Trivedi, "Robust classification and tracking of vehicles in traffic video streams," in *Proc. IEEE Conf. Intelligent Transportation Systems*, Sep. 2006, pp. 1078–1083.
[25] Z. Fu, W. Hu, and T. Tan, "Similarity based vehicle trajectory clustering and anomaly detection," in *Proc. IEEE Int'l Conf. Image Processing*, vol. 2, Sep. 2005, pp. 602–605.
[26] S. Atev, O. Masoud, R. Janardan, and N. P. Papanikolopoulos, "A collision prediction system for traffic intersections," in *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, Aug. 2005, pp. 169–174.
[27] S. Atev and N. P. Papanikolopoulos, "Multi-view vehicle tracking with constrained filter and partial measurement support," in *Proc. IEEE Int'l Conf. Robotics and Automation*, May 2008, pp. 2277–2282.
[28] J. W. Hunt and T. G. Szymanski, "A fast algorithm for computing longest common subsequences," *Communications of the ACM*, vol. 20, no. 5, pp. 350–353, May 1977.
[29] D. S. Hirschberg, "Algorithms for the longest common subsequence problem," *J. Association for Computing Machinery*, vol. 24, no. 4, pp. 664–675, Oct. 1977.
[30] M.-P. Dubuisson and A. K. Jain, "A modified Hausdorff distance for object matching," in *Proc. Int'l Conf. Pattern Recognition*, vol. 1, Oct. 1994, pp. 566–568.
[31] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, Sep. 1993.
[32] N. Meratnia and R. A. de By, "Aggregation and comparison of trajectories," in *Proc. ACM Int'l Symp. Advances in Geographic Information Systems*, Nov. 2002, pp. 49–54.
[33] M. Meilă, "Comparing clusterings – an information based distance," *J. Multivariate Analysis*, vol. 98, no. 5, pp. 837–895, May 2005.

Stefan Atev received the B.A. degree in computer science and mathematics from Luther College, Decorah IA in 2003, and the M.S. degree in computer science from the University of Minnesota, Minneapolis MN in 2007. He is a Ph.D. student at the University of Minnesota, Minneapolis MN and a senior algorithm development scientist at Vital Images, Inc, Minnetonka MN. His research interests include computer vision, machine learning, and medical image processing.

Grant Miller received the B.S. degree in computer science from the University of Minnesota, Mineapolis MN in 2009. He received an honorable mention in the 2008 Computer Research Association's Outstanding Undergraduate Award Competition. He is currently working in mobile applications, and his research interests include computer vision and programming languages.

Nikolaos P. Papanikolopoulos (F'07) received the Diploma in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 1987 and the M.S.E.E. degree in electrical engineering and the Ph.D. degree in electrical and computer engineering from CarnegieMellon University, Pittsburgh, PA, in 1988 and 1992, respectively. He is currently a Distinguished McKnight University Professor with the Department of Computer Science, University of Minnesota, and the Director of the Center for Distributed Robotics and Security in Transportation Technology Research and Applications. His research interests include computer vision, sensors for transportation applications, robotics, and control. He has authored or coauthored more than 250 journal and conference papers in the aforementioned areas. Dr. Papanikolopoulos was a finalist for the Anton Philips Award for Best Student Paper in the 1991 IEEE International Conference on Robotics and Automation and the recipient of the Best Video Award in the 2000 IEEE International Conference on Robotics and Automation. Furthermore, he was the recipient of the Kritski Fellowship in 1986 and 1987. He was a McKnight Land- Grant Professor with the University of Minnesota for the period 1995–1997 and has received the National Science Foundation (NSF) Research Initiation and Early Career Development Awards. He was also awarded the Faculty Creativity Award from the University of Minnesota. One of his papers (coauthored by O. Masoud) was awarded the IEEE Vehicular Technology Society 2001 Best Land Transportation Paper Award. Finally, he has received grants from DARPA, DHS, the U.S. Army, the U.S. Air Force, Sandia National Laboratories, NSF, Microsoft, Lockheed Martin, INEEL, USDOT, MN/DOT, Honeywell, and 3M, totalling more than $17M.

附录 B 英文原文

车辆轨迹的聚类

摘要：提出了一种适用于自动视觉系统获得的车辆轨迹聚类的方法。我们结合了两种光谱聚类方法的思想，提出了一种基于 Hausdorff 距离的轨迹相似度度量方法，并对其进行了改进，以提高其鲁棒性，并考虑到轨迹是点的有序集合这一事实。我们将所提出的方法与两种著名的轨迹聚类方法在几个真实数据集上进行了比较。
关键词：轨迹聚类；时间序列相似性度量；无监督学习

1. 引言

车辆通过交通场景的轨迹是一个信息丰富的特征，它揭示了场景的结构，提供了事件发生的线索，并允许对交通对象之间的相互作用进行推断。许多视觉算法被设计用来收集轨迹数据，或者作为主要输出，或者作为操作的副产品。

本工作的重点是提出一种无监督的车辆轨迹聚类方法具有广泛的长度变化。我们对[1]中提出的初步结果进行了扩充，并将所提出的方法与两种建立良好的轨迹聚类方法进行了比较。该方法的预期应用是作为[2]交通数据收集系统的扩展，但自动聚类在碰撞预测方面有应用，如[3]。

数据挖掘和管理领域的研究人员主要关注轨迹上的相似性度量。由于使用了相对简单的聚类和索引方案，例如聚集聚类或 k-means，这些措施的复杂性是必要的。这种方法的例子是[4]，它使用最长公共子序列(LCSS)测量来聚类车辆轨迹。[5]、[6]中实际序列的编辑距离与 LCSS 密切相关。对紧凑表示和快速检索弹道数据的需求导致了降维方法的出现，例如[7]中的分段 PCA。另一种常用的轨迹和一般时间序列的距离度量是动态时间扭曲(DTW)，以[8]为例，其中 DTW 应用于轨迹的旋转不变表示。在早期的工作中，[9]认为 lcs 优于 DTW，[10]也有同样的观点得出了同样的结论。在这两种情况下，LCSS 仅在具有大量点态异常值的数据上显著优于 DTW，而这些点态异常值被 LCSS 的距离阈值所忽略;本文所提出的方法对于离群点的存在同样具有鲁棒性，但这一点对于我们的实验并不重要，因为 DTW 的性能明显优于 LCSS。

我们的工作类似于[11]，他们在一个非常类似的任务上比较了未修改的 Hausdorff 距离、LCSS 和 DTW。此外，他们还检验了一个平面欧几里德距离和一个线性维数 re 后的欧几里德距离与 PCA 沉。最后两个距离的实验需要重新采样轨迹，这将丢失信息(在特定情况下速度信息和空间保真度)。我们特别不尝试比较需要固定维轨迹表示的方法，

因为它们需要对特定的轨迹模型或特性作出承诺。我们唯一的假设是，问题的空间方面比动态方面更重要，也就是说，车辆通过场景的路径比速度/加速度更重要。这个假设在 DTW 和 LCSS 中也是隐式的，我们将不考虑从方法中删除它所需要的修改。另一项将未修改的 Hausdorff 变量与 LCSS 进行比较的研究是[12]，该研究发现当考虑到异常值时，测量值的表现也类似。

本研究的第一个目的是比较 DTW、LCSS 和一个修正的 Hausdorff 距离在交通场景下的车辆聚类轨迹任务。我们的重点不是索引和检索，而是无监督学习。本文的第二个目的是研究问题域的谱聚类方法的性能。正如我们的结果所显示的，当处理中等大小的数据集时，没有不适当的计算负担。我们使用的聚类方法是对[13]中使用对称归一化图拉普拉斯矩阵作为亲和矩阵的方法的改进。我们没有使用固定的全局尺度，如[13]，而是采用了[14]提出的尺度选择策略，将局部尺度引入到相同的谱聚类变量中。在[15]中总结的各种方法的良好性能推动了光谱聚类的使用。我们不像[13]那样使用失真度量来优化 scale 参数，而是使用它来自动检测数据中的集群数量。最后，我们将有向 Hausdorff 距离直接纳入到[14]的局部尺度方法中，而不是首先对距离进行对称，以适应该工作的框架。

本文比较的所有方法都是基于相似性的，使用非参数轨迹表示，并且除了非参数外轨迹的动态特性都是不变的为方向。基于模型(参数化)的方法允许对不同长度和不同数量的点的轨迹进行归一化，但是由于我们要评估无模型方法，所以在本比较中没有考虑到这一点。由于我们使用了以帧速率运行的自动跟踪系统的输出，所以我们没有考虑对轨迹进行放大的方法。摘要以离散小波变换、主成分分析、离散傅里叶变换和线性预测编码系数作为时间序列聚类的归一化方法，研究了[16]中轨迹的不同表示。隐马尔科夫模型也被用来表示轨迹和时间序列，如[17]和[18]。最后，以[19]为例，用自回归模型对单变量时间序列进行了建模。

[20]演示了可以从轨迹模式的无监督学习中获益的应用程序，用于事故检测和预测系统。使用[21]提取和聚类轨迹进行事件检测。在[22]中提供了关于轨迹信息如何改进跟踪性能的相关讨论，其中以前观察到的轨迹提供了用于消除后续跟踪歧义的信息。类似地，[23]使用轨迹聚类来确定交通视频中的车道边界。边界帮助阴影检测和消除，最终也改进了后续跟踪。在[24]的工作中，轨迹信息有助于车辆分类。启用异常检测，如[25]，它使用的是未经修改的[13]的光谱聚类方法。

本文的其余部分组织如下:在第二部分，我们总结了我们的数据收集方法，并包括了 LCSS 和 DTW 的定义，以保持完整性。拟议的轨迹相似性度量是发达的第三节,和具体的聚类方法概述了在第四节。使用的数据集和性能测量是 V,部分中解释结果的描述部分 VI。我们的结论与我们的言论,在第七部分提出了未来的工作。

2. 背景

我们开始这一节通过简要描述我们的数据收集方法。使用的跟踪算法类似于我们在之前的工作中使用的[26]，[27]。本节的其余部分将用于总结比较中使用的 LCSS 和 DTW 相似性度量。我们的跟踪方法的典型输出如图 1 所示。

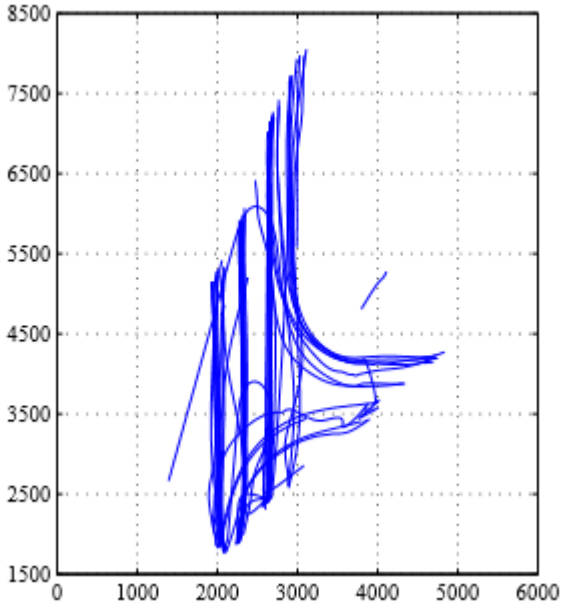


图 1 在城市交叉路口收集的样本数据集-轨迹

2.1 车辆跟踪方法

我们用彩色和黑白视频从静态交通摄像机跟踪车辆。跟踪器的基本特征是从前景中提取出来的连接组件，通过背景减法来识别，该方法将每个像素建模为含有几个组件的高斯混合。在分割之前，照度滤波和视频稳定被应用于减少伪影和错误由于相机振动和突然照度变化。图像级跟踪根据部分重叠确定前景区域从一个瞬间到另一个瞬间的关联，并使用一些启发式方法检测区域分裂和合并，这些分裂和合并可能指示动态和静态遮挡。在没有分割和合并的情况下跟踪一些视频帧的区域会被实例化为目标以进行跟踪，除非它们已经属于一个目标。第二级跟踪将目标表示为连通区域的集合。利用摄像机标定数据，将三维盒模型拟合到各目标连通区域的凸壳上。箱体模型投影在地面上的中心为车辆位置估计，箱体尺寸为车辆尺寸。虽然我们不使用这些度量来进行分类，但是我们使用这些度量来过滤掉异常值(比如被标记的行人)和错误初始化的目标。采用扩展卡尔曼滤波器来处理由于透视投影引起的非线性问题。此外，一旦确定目标已经离开场景，我们使用随机滤波器的中间输出作为卡尔曼平滑的输入，从而确保每个时间步的位置和大小估计都以所有可用的测量值为条件。从图 1 可以看出，恢复的轨迹非常平滑，表现良好。最后，由于跟踪器以帧速率工作，轨迹的采样速率为 30Hz，这就消除了插值的需

要。

2.2 LCSS 和 DTW 的相似性度量

我们使用的 LCSS 的版本是在[4]中定义的。Given 两个标量序列，$\mathbf{a} = \{a_i\}_{i=1}^{n}$和$\mathbf{b} = \{b_j\}_{j=1}^{m}$一些 $\ell_{\epsilon,\delta}(\mathbf{a},\mathbf{b})$被定义为 follows: LCSS 相似之处：

$$\ell_{\epsilon,\delta}(\mathbf{a},\mathbf{b}) = \begin{cases} 0, \ n = 0 \vee m = 0 \\ 1 + \ell_{\epsilon,\delta}(\mathbf{a}',\mathbf{b}'), \ |a_n - b_m| < \epsilon \wedge |n - m| \leq \delta \\ \max\{\ell_{\epsilon,\delta}(\mathbf{a}',\mathbf{b}), \ell_{\epsilon,\delta}(\mathbf{a},\mathbf{b}')\}, \ \text{otherwise}, \end{cases} \tag{1}$$

3. 轨迹聚类

在本节中，我们将描述所使用的两种聚类方法，并阐明我们用来确定由第二节跟踪方法产生的轨迹中哪些被认为是离群值的规则。在接下来的阐述中，dij 表示第 i 条和第 j 条输入轨迹之间的距离(采用凝聚法)，kij 表示经过适当的尺度选择后两者的相似度(采用谱法)。

3.1 离群值过滤

由于视觉跟踪在各种情况下都可能失败，并非所有自动收集的轨迹都对应于一辆车，在极端情况下，被跟踪的对象根本不是车辆。这对于交通交叉口尤其适用，因为在中等密度的交通中，由于交通阻塞，跟踪非常困难。这样的异常轨迹，如果在输入中加入了聚类算法，产生了许多单例聚类，阻碍了学习过程。我们会尽可能提前移除异常值。由于我们的跟踪方法产生尺寸估计和真实单位的位置，我们有一个合理的方法来预过滤大多数轨迹产生跟踪失败或合并的目标。

如果被跟踪物体的轨迹违反了以下任何一条规则，则被认为是离群值，在我们的实验中不使用:

1) 至少有 90 个样本(高速公路为 30 个)可用。
2) 对象长度必须超过宽度。
3) 物体长度必须在 1.5 - 6 米之间。
4) 宽度必须在 1 到 4 米之间。
5) 轨迹上的最大瞬时速度必须在每小时 10 至 100 公里之间。
6) 物体的转向速度必须低于 45◦每帧。

一定数量的符合上述所有标准的轨迹被认为是 ground-truth 中的异常值。我们将这些异常值保留在算法的输入中，以测试这些方法的鲁棒性。这与[11]形成了对比，[11]在

他们的实验中手动消除了所有的异常值。由于没有任何一种聚类方法能够显式地检测离群值，因此我们将离群值轨迹从聚类混淆矩阵中排除(也就是说，算法收到的分数不受分配给离群值的聚类的影响)。

3.2 凝聚聚类

我们使用的凝聚聚类方法与[4]集群中的方法相同。最近的两个集群迭代合并,直到所有集群相距一段距离大于预定阈值 τ 。如果 I 和 J 表示两个不相交的团簇轨迹的指数集，则团簇之间的距离为:

$$d_{\text{avg}}(I,J) = \frac{1}{|I|\,|J|}\sum_{i\in I}\sum_{j\in J} d_{ij}.$$

在我们的测试中,我们指定所需的数量的集群,而不是选择一个值 τ 由于集群的数量从手动真实数据。的范围值 τ 生产相同数量的集群是指示性的敏感性底层距离测量,并将我们的实验的一部分。

3.3 谱聚类

如前所述，我们使用了[13]提出的聚类算法的一个变体，但是使用了[14]的局部缩放方法。为了完整起见，我们在这里包括了对算法的描述。注意，我们使用[13]中使用的失真度量自动选择集群的数量，以选择最佳的全局规模。

光谱聚类方法的输入为元素 kij 为 1⩽i,j⩽n 的亲和矩阵 K，该过程的第一步是对 K 的行和列进行归一化，得到归一化的亲和度 L:

$$\mathbf{L} = \mathbf{W}^{-1/2}\mathbf{K}\mathbf{W}^{-1/2},$$

3.4 聚类实验

在可能的范围内，我们在所有相似性度量中使用等效参数。LCSS 特定问题在于 一些选择的事实, 算法将无法产生预期的数量的集群, 因为太多的轨迹在无限的距离彼此, 但 LCSS 距离的任意两个轨迹隔开距离超过 ε 到处导致∞, 我们显示的结果的算法使用两个不同的设置，第一个是 12 英尺(the lane-width scenes), 和第二个是最小的值足够大, 这样. 所需 数量 的 集群在自动检测到的集群数量与实际集群数量不同的情况下，我们给出了该算法的两个得分。为了说明所提出的轨迹距离的适用性，我们还利用改进的 Hausdorff 距离进行聚类。对于修改后的豪斯多夫距离与 lcs,我们使用一半的松弛参数轨迹长度,也就是说,我们集 w = 0.5 为修改后的距离和 δ = 0.5 lcs(非常类似于[4]的值)。最

后,参数 α 的分离方法设置为 0.88,这是决心是最优的数据集的一个子集 1 [1]。注意,最优选择 α 在[1]无追索权 groundtruth 数据,这是最小化畸变分数的价值在第四部分讨论。相同的值被用于所有其他数据集具有良好的优化结果并不是每个剩余的数据集。

4. 结果

各种聚类方法与 ground-truth 之间的信息变化如表 2 所示。对于产生与 ground truth 相同数量集群的集群，还显示了集群错误。集群的实际数量输入是已知的所有算法。LCSS-based 方法表明 ε，因为它多种多样的值。每个距离度量的最佳结果(通过 dVI 度量)以斜体突出显示，而整个最佳结果以粗体突出显示。表二的方法名称如下:

Name	Distance	Type	Setting of k
HA	Hausdorff	Agglomerative	Manual
HS	Hausdorff	Spectral	Automatic
HM	Hausdorff	Spectral	Manual
DA	DTW	Agglomerative	Manual
DS	DTW	Spectral	Automatic
DM	DTW	Spectral	Manual
LA-ϵ	LCSS	Agglomerative	Manual
LS-ϵ	LCSS	Spectral	Automatic
LM-ϵ	LCSS	Spectral	Manual

4.1 讨论

表二的第一个非常明确的结果是，改进的 Hausdorff 距离更好地反映了轨迹相似的概念;即使以聚集的方式使用，这个距离也比 DTW 和 LCSS 更好。改进后的 Hausdorff 距离与谱聚类算法相比也有更好的性能。第二个观察结果是，当给定正确的聚类数目时，光谱方法在 LCSS 和 Hausdorff 距离下的表现优于等价凝聚法。使用较大阈值的 LCSS 距离并不能从光谱聚类中受益，这是使用不合理的高阈值的结果(这是必要的，因此聚集方法产生所需的聚类数量)。光谱聚类的优点在数据集 3 中尤其明显，其中由于成功跟踪的起始点差异很大，因此集群包含许多部分重叠的轨迹。虽然数据集 3 看起来非常简单，但是轨迹长度和轨迹端点的巨大变异性挑战了所有聚集聚类变量，因为簇间和簇内距离都有很大的方差。高速公路的数据集根本没有挑战光谱方法——集群的数量和类别的成员都被完美地恢复了。令人惊讶的是，HS 在第二个数据集上的表现优于 HM，尽管它低估了集群的真实数量(比较图 5b 和图 6b)。这可以解释为，HS 合并了两个有明显重叠的星团，但在其他方面产生的结果与所有其他轨迹的基本事实相同。由于重叠的原因，HM 也合并了相同的集群，但为了获得指定的 6 个集群，它将另一个稀疏集群拆分为两个子集群。

当以聚集方式使用 LCSS 时，它在这个问题上的性能令人沮丧。给定一个非常 合理的 阈值, LA-12 未能足够的簇合并到指定的 clusters.数量正确问题是 LCSS 只产生了两种类型的距离——非常接近于最小可能值和无穷，任何离群值或分离良好的集群都会产生许多无限的距离。Increasing 改进 性能 的 方法 对 高速 公路 数据 sets, 但 相反的 效果 在 十字 路口 数据 sets, 和 消极 的 谱 聚 类 结果 的 影响虽然 DTW 的性能不是很好，但我们认为这主要是由于(4)和(5)非常严格的约束，需要匹配轨迹的起始点。由于跟踪器中目标初始化的不可预测性和透视的影响，同一簇内的轨迹可能具有不同的起始位置和结束位置。由于 Hausdorff 距离并没有施加这样的约束，因此在数据集 3 上它比 DTW 有明显的优势，而在数据集 4 上则有较小的优势。我们穷人的结果感到很惊讶当使用 DTW 在光谱框架中,并没有明确的解释观察,除了要注意一点:在数据集的最大 intra-cluster 轨迹长度的变化,数据集 1 DS 和 DM 表现不好,而相同的方法优于 DA 的数据集。

4.2 方法时间消耗

影响比较方法运行时间的主要因素有两个:一对距离矩阵的计算和实际的聚类。对于所有比较的方法，第一步的渐近复杂性是非常相似的;在实际应用中，如表 3 所示，改进后的 Hausdorff 距离和 lcs 的运行时间非常相似，DTW 慢了一个常数倍。DTW 的长运行时的原因是,我们不使用限带版本的距离(松弛参数 δ 为 w 与 lcs 提出距离减少运行时间)。这个组件的所有实现都是在 c++中实现的，但是没有进行大量的优化。光谱聚类和凝聚聚类的运行时间比较接近。由于不同的原因，这两个时期都被夸大了。在光谱聚类的情况下，我们计算

K 的所有特征值和特征向量，尽管通常我们只需要最上面的几个。这是 Matlab 中稠密特征解的一个局限性。聚集聚类也在 Matlab 中实现，但是很少使用内置函数(如 eig 和 kmeans)，因此在性能上处于劣势。

TABLE III: Running time of the methods in seconds.

Data Set (Size)	Method	Similarity	Clustering	Total
1 (403)	HS	13.8	1.19	14.99
	HA	13.8	1.03	14.83
	LA-12	19.7	2.26	21.96
	DA	269.9	1.03	270.93
2 (391)	HS	1.44	0.62	2.06
	HA	1.44	0.91	2.35
	LA-12	1.56	2.08	3.64
	DA	20.6	0.92	21.52
3 (161)	HS	0.89	0.075	0.965
	HA	0.89	0.039	0.929
	LA-12	0.97	0.048	1.018
	DA	13.8	0.038	13.838
4 (240)	HS	0.81	0.18	0.99
	HA	0.81	0.10	0.91
	LA-12	0.88	0.27	1.15
	DA	11.6	0.10	11.7

4.3 凝聚聚类的敏感性

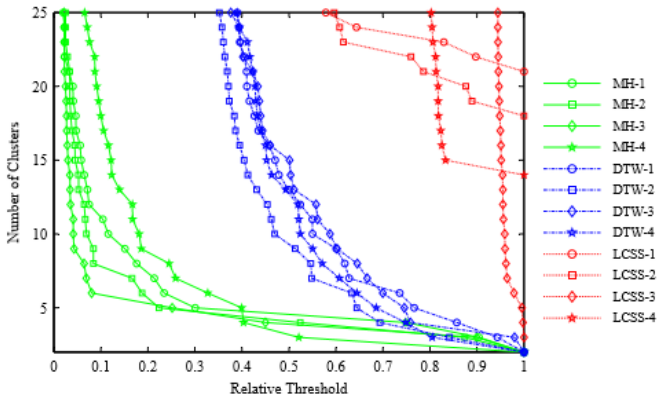


图 2 聚集聚类中的阈值与生成的聚类数量的关系图

图 7 说明了不同距离度量下的聚集聚类行为。步骤的宽度表示距离阈值可以在不改变集群数量的情况下改变的范围。DTW 和改进的 Hausdorff 距离的表现都与预期一致——随着集群数量的减少，阈值范围增大。改进的 Hausdorff 距离在需要少量簇时对阈值不敏感，但在阈值较小时比 DTW 更敏感。LCSS 距离的不稳定行为是由于返回的距离要么非常小，要么无穷大。图显示，LCSS 无法合并超过给定点的集群，因为所有距离都变得无穷大。请注意，修改后的 Hausdorff 距离图的陡峭部分出现在四个数据集的每个数据集达到正确的聚类数目之前。这意味着通过搜索簇大小与阈值图上的拐点，利用 Hausdorff 距离可以更容易地自动检测出正确的簇数。

5. 结论和未来展望

提出了一种基于定向 Hausdorff 距离的轨迹相似度度量方法，并将其应用于光谱聚类方法和聚类方法。轨迹之间的一些对应的实施结构修改必要让豪斯多夫距离适用于有

序点集的比较,而拒绝一定量的最差的比赛距离提高其鲁棒性噪音,增加了部分匹配的宽容。我们没有像[13]建议的那样使用失真度量来选择全局最优的尺度参数，而是使用该度量来确定一个集合的最优集群数量

通过检查亲和矩阵的谱得到的可能性。从[14]中选择的局部尺度对于轨迹的聚类效果非常好，因此不需要搜索任何尺度参数。光谱方法的一个可能的未来改进是允许离群值检测。将 DTW 距离的严格对应与所提出的 Hausdorff 距离的思想结合起来应该是可能的，因为匹配的严格单调性似乎更适合于时间序列。

一般来说，DTW 使用的对应点对的平均距离似乎比我们使用的鲁棒裁剪最大值更容易受到异常值的影响。DTW 中起始点/结束点约束的放宽也可以提高不同长度轨迹匹配距离的适用性，并允许部分匹配。但是，(6)的单调性约束和(8)的裁剪极大值是否可以用复杂度 O(mn)的方法结合起来比较 n-元素和 m-元素的轨迹，这一点还不清楚。我们打算将来调查这个问题。

基于 LCSS 的距离测量在更简单的数据集上的表现也令人失望。大多数情况下，轨迹被认为非常相似或非常不相似，没有明显的中间情况。这将使 LCSS 距离不适用于具有局部尺度的谱聚类，因为相对距离将类似地在一些极值处聚类。使用一些理想的质量的 LCSS 是可以模仿的。