

基于哈希和路网边权的地图匹配算法及其应用

邵天浩,张宏军,程 恺,唐新德,莫 斐,张 可

(陆军工程大学 指挥控制工程学院,江苏 南京 210000)

摘 要:现如今车载导航的使用越来越普及,海量的车辆行驶数据使得车辆驾驶行为分析成为一项新的研究热点,但是由于车载导航采集的行驶数据存在一定的偏差,需要对其进行有效的预处理以便准确分析驾驶行为。提出一种以确定性地图匹配算法为依据的新型地图匹配算法,该算法以邻接表的形式存储路网信息,利用 geohash 函数筛选数据点附近的道路,在匹配过程中通过添加滑动窗口来提高正确率,最后在修正路网边权的基础上搜索最佳路线,将其作为填充空白数据点的依据。该算法便于实现,相比于传统的确定性地图匹配算法具有更高的匹配精度,相对于不确定性地图匹配算法具有更高的匹配速度,并且适用于大段数据异常的情况,可以提高车辆行驶数据的质量,同时满足了车辆实时定位的处理需求。

关键词:地图匹配算法;邻接表;geohash 函数;路网边权修正;缺失值填充

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2020)08-0140-07

doi:10.3969/j.issn.1673-629X.2020.08.024

Map Matching Algorithm Based on Hash and Edge Weight of Road Network and Its Application

SHAO Tian-hao, ZHANG Hong-jun, CHENG Kai, TANG Xin-de, MO Fei, ZHANG Ke

(School of Command and Control Engineering, Army Engineering University of PLA, Nanjing 210000, China)

Abstract: Nowadays, vehicle navigation is becoming more and more popular. Massive vehicle driving data makes vehicle driving behavior analysis a new research hotspot. However, due to some deviations in the collected driving data, effective preprocessing of the driving data is required to accurately analyze the driving behaviors. Based on the deterministic map matching algorithm, we propose a new map matching algorithm. Firstly, this algorithm uses adjacency table to store the road network information. Secondly, by adding a sliding window to improve the accuracy in the matching process, geohash function is used to filter the ways near the data points. Finally, the best route is searched on the basis of correcting the edge weight of road network, and this best route is the basis of filling blank data points. Easily to implement, this algorithm has higher matching accuracy compared with the traditional deterministic map matching algorithm, and it has higher matching speed than the uncertainty of map matching algorithm. Besides, it can not only improve the quality of traffic data, but also meet the processing requirements of vehicle real-time positioning with the fitness of dealing with large data exception.

Key words: map matching algorithm; adjacency table; geohash function; correcting edge weight of road network; missing data imputation

0 引 言

近年来,随着车载导航的普及率越来越高,海量的车辆行驶数据被导航收集并上传至车联网系统。这些数据可以体现车辆在行驶过程中的各种信息,如经纬度、方位角、GPS 速度、行驶里程等。如何从海量的行驶数据中分析出车辆驾驶行为的特征,将数据转化为价值,是当前研究的一个热点问题^[1-2]。但是由于车载导航自身的局限性以及数据采集的不稳定性,采集到的行驶数据往往质量不高,这会导致其分析出的结

果与实际情况有所偏差。因此,对车辆行驶数据进行有效的预处理以提高其质量,是后续数据分析的重要前提。

通过地图匹配来修正车辆行驶数据是如今最常用的方法,目前地图匹配算法主要有确定性地图匹配算法和不确定性地图匹配算法两种^[3]。

确定性地图匹配算法是指匹配的结果是确定而非随机的,主要通过几何方式进行匹配,包括点到点、点到弧、弧到弧三种方式^[4]。它考虑了车辆方位角与道

收稿日期:2019-09-27

修回日期:2020-01-20

基金项目:国家自然科学基金(61806221)

作者简介:邵天浩(1996-),男,硕士研究生,研究方向为数据挖掘;张宏军,博导,教授,从事软件工程、机器学习等研究。

路方向的相似性、采样点与道路的距离以及车辆行驶速度等多个方面,将各要素加权之后选择最合适的匹配路段。这类算法原理简单,实现容易,但其稳定性不高,尤其是在选择初始匹配路段时没有进行初步筛选^[5],导致匹配的精度下降。

不确定性地图匹配算法是指相同数据点多次匹配的结果是非确定的,依据一定概率进行匹配,主要分为概率统计算法和模糊逻辑算法两种。概率统计算法主要依据采样点信息设置置信区域,利用概率准则来确定需要搜索的道路阈值,在匹配过程中需要通过复杂的数学公式来推导定位误差椭圆,这一方法虽然具有较高的匹配精度,但大量的推导使其匹配速度下降^[3]。模糊逻辑算法与传统的模糊推理系统类似,通过模糊化、模糊推理机和去模糊化三个部分搭配隶属度函数在推理过程中进行道路匹配,这一算法的实现虽然较概率统计算法来说容易,但其模糊推理机中参数的设定主要依靠专家知识,缺少一定的推理支撑^[6]。

总的来说,相对容易实现的匹配算法其匹配精度较低,匹配精度高的算法计算复杂,匹配速度较慢。同时,上述匹配算法均没有提出存在大段异常点时的匹配模式。为了兼顾高效与匹配速度,文中在确定性地图匹配算法的基础上,提出了一种新型地图匹配算法。首先将路网信息以邻接表形式存储,然后通过 geohash 函数来筛选数据点周围的道路,并添加一个滑动窗口来减小匹配错误率,最后通过修正路网边权来填充空白数据点。该算法实现简单,较传统的地图匹配算法有更高的匹配精度,较概率统计算法和模糊逻辑算法有更快的匹配速度,并且在数据存在大段异常时也能很好工作,匹配结果提高了车辆行驶数据的质量,可以推广到车辆的实时定位。

1 算法的设计与实现

基于哈希函数与路网边权修正的地图匹配算法的具体流程如图1所示^[7]。

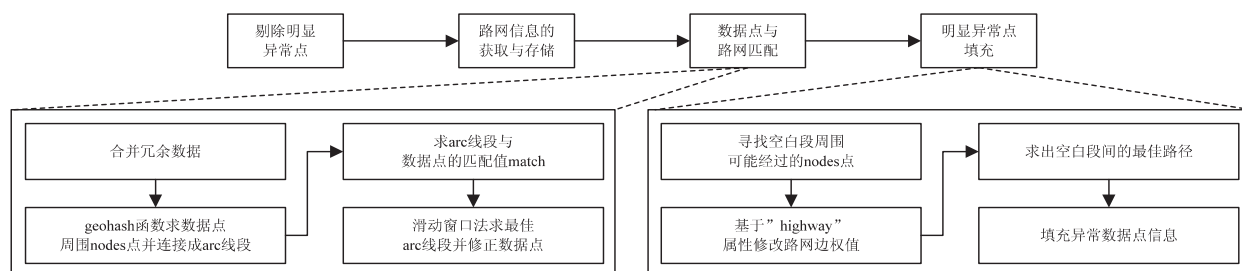


图1 地图匹配算法流程

主要包括剔除明显异常点、路网信息的获取与存储、数据点与路网匹配、明显异常点填充四个步骤,其中数据点与路网匹配、明显异常点填充又包含相应的子流程。下面分别阐述每个步骤及子流程的设计实现方法。

1.1 剔除明显异常点

由于GPS信号有时候存在不稳定性,车辆行驶数据中包含了大量的漂移数据,这些数据与正常的数据偏差很大,是明显的异常点,首先需要对这样的明显异常点进行剔除。

信号漂移会导致最后一个正常的数据点和第一个异常的数据点之间有较大的经纬度差异,可以根据这一差异对数据集中明显的异常点进行剔除^[8]。

已知地表上两点的经纬度,可以得出两点间的最短距离 D_e ,如式(1)所示。

$$D_e = R \cdot \arccos(\cos x_a \cos y_a \cos(y_a - x_a) + \sin x_a \sin y_a) \quad (1)$$

根据两个数据点 i 、 j 采集到的经纬度数据,可以求出两个数据点之间的平均速度,并用平均速度代替后一个数据点的瞬时速度,如式(2)所示,其中 D_e 为两数据点的经纬度距离。

$$V_b = \overline{V_{ab}} = \frac{D_e}{\Delta t_{ab}} \quad (2)$$

若某一数据点 i 的经纬度速度与GPS速度满足式(3),则认为其是正常数据点,否则将其剔除,选择下一数据点继续判断。

$$V_c < |V_i - V_c| \times 30\% \quad (3)$$

利用上述过程对明显错误的异常点进行剔除的方式较为粗糙,会留下一些异常点没有剔除干净,但这种方法的好处是可以处理大段的连续异常数据,这是利用隐马尔可夫模型^[9]来进行预处理所不能达到的,因为过长的隐马尔可夫链会失去预测的准确性。未被剔除的不太明显异常点可以通过道路匹配的方式进行修正。同时,针对明显异常点剔除之后的大片缺失数据,可以通过修正路网边权来进行填补。

1.2 路网信息的获取与存储

想要将数据点与道路进行匹配,从而修正数据点的值,首先需要获取路网信息,获取的信息主要包括地图上的每条道路和每个点,以及它们之间的内在联系。

将路网信息中的道路信息和交叉点信息进行提取,分别用ways道路和nodes点来表示,对两者进行简化,留下各自的唯一标识id、经纬度坐标以及其关联

关系。最后用邻接表来存储化简之后的信息,邻接表中的顶点对应 nodes 点,有向边对应两个 nodes 点之间有一条单向的道路。

式(4)表示邻接表存储的数据结构,其中 id 表示某一数据点的唯一标识 id ,该数据点与附近 n 个数据点相邻; id_n 表示与该数据点相邻的第 n 个数据点; wid_n 表示对应两数据点连接而成的道路的唯一标识符 id ; d_n 表示对应两数据点之间的距离; a_n 表示对应两数据点连接而成的道路的方位角,以正北方向为基准线。

$$id: [id_1, wid_1, d_1, a_1; \dots; id_n, wid_n, d_n, a_n] \quad (4)$$

需要注意的是,在实际应用中,为了减少搜索空间,会先确定车辆轨迹的经纬度范围矩阵,根据范围矩阵来生成邻接表,从而减少数据点匹配所用时间。

1.3 数据点与道路匹配

剔除了大部分异常点之后行驶数据在地图上表示为多段行驶轨迹,每行驶轨迹之间缺失的数据即是被剔除掉的明显异常点。需要将每段行驶数据进行更进一步的处理,使其具有一定的规律性,处理依据如下:

如果相邻两个数据点之间的采样间隔超过 2 分钟,则认为两者属于不同的行驶段;如果某一段行驶段的起始点和终点的采样间隔小于 2 分钟,则抛弃这一行驶段。

对数据分段之后,可以将每个数据点与附近的道路进行匹配,并且修正数据点。可以按照如下步骤来完成,以每个数据点作为基本操作单位:

Step1: 利用 geohash 函数求出每个数据点周围的 nodes 点,并将找到的 nodes 点通过其与 ways 道路的关系连接成线段 arc;

Step2: 求出数据点与 arc 线段的距离 distance 和夹角 angle,依据数据点的行驶速度对两者进行加权,求出每个数据点与周围 arc 线段的匹配值 match;

Step3: 依照一定的筛选规则,选出与每个数据点最匹配的 arc 线段,使其与上一数据点匹配的 arc 线段相连,并修正数据点的属性值;

Step4: 对下一个数据点计算 Step1,直到该行驶段的数据点计算完为止。

其中 Step1 ~ Step3 的具体实现过程如下。

1.3.1 筛选数据点周围道路

由于数据点与路网信息的邻接表之间没有直接的关联关系,因此,若需要寻找数据点周围的 nodes 点是十分困难的,需要遍历整个邻接表,通过计算数据点与 nodes 点之间的距离来筛选符合条件的 nodes 点。虽然已经通过范围矩阵将邻接表中的数据进行初选,但逐条计算的方式仍然要耗费大量时间,因此在寻找数据点周围的 nodes 点时,文中利用了空间搜索算法中

的 geohash 函数。

geohash 函数的基本思想是将地球根据经纬度看成一个二维平面,将二维的经纬度转换成字符串编码^[10],例如,图 2 展示了某一地区中 9 个区域的 geohash 编码,每一个 5 位字符串编码代表了一块矩形区域。而不同的编码长度表示不同的范围区间,字符串越长,表示的范围越精确。geohash 的前 5 位编码精度大约在 2 km,文中使用的算法需要筛选每个点周围 500 m 内的 nodes 点,因此 5 位 geohash 编码可以满足条件。



图 2 geohash 编码示意图

利用 geohash 函数求出每个 nodes 点的 geohash 值,通过字符串匹配的方式,寻找与数据点的 geohash 值前 5 位相同的 nodes 点。需要注意的是,虽然前 5 位 geohash 值匹配的精度在 2 km,但如果数据点的位置靠近划分矩形的边缘,那么某些距离数据点 500 m 内的 nodes 点也不在矩阵内。因此,在经过 geohash 函数计算之后,还需要基于 nodes 值对数据进行进一步拓展筛选。对于数据点所在矩阵内的每个 nodes 点,拓展寻找其 500 m 范围内的 nodes 点,寻找的方法是基于给定 Limit 的深度优先搜索。邻接表的深度优先搜索可以基于栈结构快速进行,当搜索深度大于 500 m 时即停止该分支的搜索进行回溯。这样就对原有矩阵进行了拓展,保证了原本距离数据点 500 m 内但不在矩阵内的点也在拓展之后的 nodes 点内。

最后,遍历所有拓展后得到的 nodes 点,筛选出数据点 500 m 范围内的 nodes 点,并将其依据邻接表的信息进行连接,得到基于 ways 数据的线段,将其称为 arc 线段。

1.3.2 计算道路与数据点的匹配度

对于每一个数据点,已经得到其周围 500 m 范围内的 arc 线段,对数据点与 arc 线段之间的距离和夹角进行加权,计算数据点与每个 arc 线段的匹配值 match,该值越大,说明数据点与道路越匹配。两者的

距离 distance 用点到线段的最小距离表示, 夹角 angle 为点的方位角与线段的夹角, 两者的值越小, 说明数据点与道路越匹配, 匹配度应该越高, 因此它们之间有负相关的关系。

得到 distance 与 angle 之后, 首先对其进行归一化, 然后利用式(5)计算数据点到每个 arc 线段的 match 值^[11]:

$$\text{match} = 1 - [(1 - \lambda) \times \text{distance} + \lambda \times \text{angle}] \quad (5)$$

其中, λ 表示 angle 指标的权重, λ 的值与数据点的 GPS 速度有关: 当数据点的速度小于 1 m/s 时 λ 的值为 0, 速度大于 10 m/s 时 λ 的值为 0.5, 速度介于两者之间时 λ 的值按线性从 0 到 0.5 增长^[12]。假设 distance 指标的权重永远大于 0.5, 这是因为本文认为距离指标的重要性大于夹角指标, 因为数据点需要选择更相邻的道路进行匹配。

1.3.3 求最佳匹配道路

有了数据点与 arc 线段的匹配值 match, 就可以将两者进行匹配, 这一过程主要分为三步。

首先, 根据 distance 对 arc 线段进行初选, 依据为 arc 线段中“highway”属性的值, 选出距离数据点 100 m 内的所有道路, 距离数据点 200 m 内的所有主要道路, 距离数据点 400 m 内的所有干线道路, 距离数据点 500 m 内的所有高速公路。

其次, 将初选之后 arc 线段依 match 值从大到小排序, 选择 match 值最大的 5 条 arc 线段^[13], 作为最终匹配的候选线段。若需要匹配的数据点为行驶段的起始点, 则选择 match 值最大的那条线段作为匹配的道路; 若需要匹配的数据点为非起始点, 需要判断 5 条 arc 线段是否满足以下两个条件之一^[14]:

(1) 该 arc 线段与上一数据点匹配的线段是否重复;

(2) 该 arc 线段与上一数据点匹配的线段是否相连。

若 5 条 arc 线段中存在某条线段满足上述两个条件之一, 则认为这条线段是与该数据点匹配的 arc 线段, 将其与前一数据点匹配的 arc 线段相连, 将其认为是车辆的真实行驶轨迹。

若不存在 arc 线段满足上述条件中的任意一个, 则计算上一数据点匹配的 arc 线段尾端点到 5 条 arc 线段首端点的最短路径, 这一最短路径并非直线最短, 而是在邻接表中通过可达边的最短路径。最终选择最短路径最小的 arc 线段作为该数据点匹配的 arc 线段, 将前一数据点匹配的 arc 线段经过最短路径上的 nodes 点与其连接, 将其认为是车辆的真实行驶轨迹。

需要注意的是, 这一方法的基本思想是贪婪的, 即

倘若在后期发现某条 arc 线段完全不符合要求, 也无法改变前面的匹配结果。为了避免这种情况, 文中在实际求解中利用一个滑动窗口, 该窗口保留当前匹配数据点前 500 m 里程内所有数据点的 5 条最佳 arc 线段匹配列表, 若发现当前数据点的 5 条 arc 线段均无法匹配, 则说明匹配出错, 应该向前回溯至上个数据点, 选择下一条最合适的 arc 线段重新匹配。

完成数据点与附近的路网信息匹配之后, 倘若数据点不在所匹配的道路上, 则需要利用投影法^[15]对数据点的经纬度信息进行修正, 将数据点在道路上的垂足(垂足在 arc 线段上)或道路中点(垂足不在 arc 线段上)作为修正的新数据点。

1.4 明显异常点填充

1.3 节将数据点与路网信息结合, 通过路网信息对异常的数据点进行了有效的修正, 但修正的数据点并不包含 1.1 节中被剔除的明显异常的数据点, 这使得修正完之后的数据存在大段缺失, 导致在车辆行驶轨迹中存在一定的空白段。

为了填充各个行驶段之间的空白数据, 文中提出通过修正路网边权来寻找空白段之间最合适的填补路径, 寻找步骤如下:

Step1: 寻找空白数据段可能经过的 nodes 点以及 nodes 点连接成的 ways 道路;

Step2: 按照“highway”属性修改邻接表中属性“d”的值;

Step3: 求出空白段之间的最佳路径;

Step4: 依据最佳路径填充空白数据点。

Step1 ~ Step4 的实现过程如下:

在填充明显异常点之前, 首先选择空白数据段前后两段数据的尾部和头部的 nodes 点, 以前段数据尾部的 nodes 点为圆心, 以 1.3.1 节中方法来寻找两个 nodes 点之间距离 1.5 倍范围内的所有 nodes 点。将符合条件的所有 nodes 点仍然以邻接表的形式存储, 但必须将两点之间的经纬度距离进行加权。

1.4.1 修改邻接表“d”属性

由于现实中公路的建设并非两点之间的最短路径, 因此在填充两段行驶段之间空白时不能简单地通过寻找最短路径; 并且考虑到车辆运输过程中很少会跨越多等级的公路行驶, 即从高速公路直接行驶到县区道路的情况是很少的, 只有在开头与结尾才会出现, 因此可以按照“highway”属性的关系来对邻接表中“d”属性进行修改。

想要对邻接表中的属性“d”进行修改, 首先需要知道空白数据前后两段数据点所匹配道路的“highway”属性, 若需要加权的 ways 道路其“highway”属性与前后两段的“highway”属性相同或

者相近,则赋予较小的权重,这样在搜索时可以更优先选择这一道路。同时依据需要加权的 ways 道路中点与前后两段数据点的直线距离,重新分配前后两段数据点对其的权重分配。

根据“highway”属性赋予权重的依据如表 1 所示。

表 1 highway 属性权重

highway	motorway	trunk	primary	secondary	other
motorway	0.0	0.5	1.0	1.5	2.0
trunk	0.5	0.0	0.5	1.0	1.5
primary	1.0	0.5	0.0	0.5	1.0
secondary	1.5	1.0	0.5	0.0	0.5
other	2.0	1.5	1.0	0.5	0.0

表 1 中第一列表示前后两段数据点所匹配道路的“highway”属性,第一行表示需要加权的 ways 道路的“highway”属性。根据式(6)重新分配权重方式。

$$\omega = \frac{d_2}{d_1 + d_2} \times \omega_1 + \frac{d_1}{d_1 + d_2} \times \omega_2 \quad (6)$$

其中, ω_1 、 ω_2 表示前后两数据段赋予的权重, d_1 、 d_2 表示 ways 道路中点距离前后两数据段端点的直线距离。

图 3 示例表示了实际中的权重确定方式。

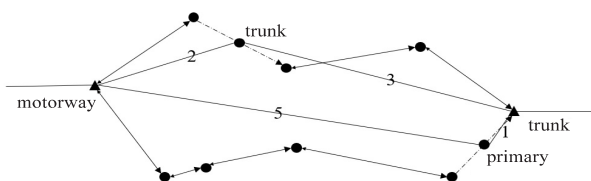


图 3 权重修改示意图

图 3 中空白数据前后两段端点用三角形表示,前段数据匹配的道路属性为 motorway,后端数据匹配的道路属性为 trunk。上侧虚线 ways 道路属性为 trunk,下侧虚线 ways 道路属性为 primary。上侧虚线中点到前后端点的直线距离分别为 2 和 3;下侧虚线中点到前后端点的直线距离分别为 5 和 1。

若要求图中虚线所示两段道路的权重,结合属性和表 1,得出上侧虚线道路的 $\omega_1 = 0.5$ 、 $\omega_2 = 0$,又因为 $d_1 = 2$ 、 $d_2 = 3$,因此求出上侧虚线道路的权重:

$$\omega = \frac{3}{2 + 3} \times 0.5 + \frac{2}{2 + 3} \times 0 = 0.3$$

同理,上侧虚线道路的 $\omega_1 = 1.0$ 、 $\omega_2 = 0.5$,又因为 $d_1 = 5$ 、 $d_2 = 1$,因此求出上侧虚线道路的权重:

$$\omega = \frac{1}{5 + 1} \times 1 + \frac{5}{5 + 1} \times 0.5 = 0.58$$

根据上述方式求出每条道路的权重,然后用权重乘以原始数据,即得到修改之后“d”属性内的值 d’。

1.4.2 填充异常数据点信息

对邻接表中属性“d”进行加权修改之后,利用传

统的 Dijkstra 算法或者 Floyd 算法搜索得到空白段之间基于权重的最佳路径。将最佳路径通过的 nodes 点连接起来,认为是车辆实际的行驶轨迹。

有了车辆的行驶轨迹,仍需要在轨迹上寻找数据采样定位点,最终完成填充空白段之间的数据。将在 1.1 节中被剔除的所有异常数据点连成一条直线,算出每个异常点在整条直线上的比例,在空白段的相同比例处添加相应数据点,这样保证了重新填充的正常点个数与剔除的异常点个数相等。

对于新添加的数据点,仅存在经度与纬度两个属性的值,需要获取对应原始数据点中的方位角、GPS 速度和行驶里程等信息,作为新添加数据点对应属性的值。

2 实验验证

文中以某公司运输车辆的 10 份行驶数据作为实验数据集,数据集包括车号、方位角、经纬、纬度、GPS 速度和里程 6 个属性值。

从网站 OpenStreetMapnodes 上下载开源地图数据,该网站提供的地图数据是以 xml 格式存储的,将全球的地图数据保存为一个大型的 xml 文件,并用特定的属性来描述存储的数据。

提取地图数据中属性为“nodes”的节点信息和“ways”的道路信息。“nodes”表示地图上的节点,包括唯一标识符 id 和经纬度坐标 lng 与 lat,其属性中的“ways”表示该点落于这条道路上;“ways”表示地图上的道路,包括唯一标识符 id 和道路属性 highway,其属性中的“nodes”表示该道路所经过的点。根据式(4)的邻接表存储格式,将“nodes”信息和“ways”信息转化为邻接表存储的一个实例的表示如下:

052:[756,131,128.10,6.67;045,500,244.37,186.81]

其中 052 表示某一 nodes 点的 id,该 nodes 点与 2 个附近 nodes 点相邻,756 和 045 分别为两个相邻点的 id;131 和 500 表示原数据点和两个相邻数据点连接而成的道路 id;128.10 和 244.37 表示原数据点和两个相邻数据点之间的距离;6.67 和 186.81 表示原数据点和两个相邻数据点连接而成的道路的方位角。

以 1 号车为例,将未经数据预处理的车辆行驶轨迹绘制在图 4 所示的左侧地图上。

由图 4 左侧地图可以看出,原始的车辆行驶数据由于具有较多异常点而导致轨迹难以辨认。将剔除明显异常点之后的数据集与道路进行匹配,并且在匹配的过程中修正和填充数据点的值,最终修正完成之后的车辆行驶轨迹如图 4 右侧地图所示,其中,不同灰度的轨迹表示不同的行驶阶段。

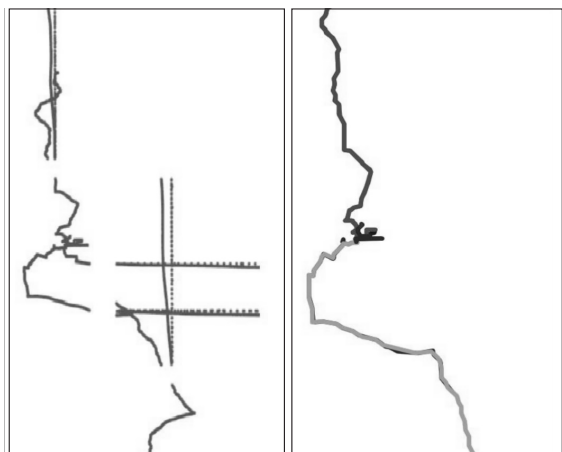


图4 1号车初始行驶轨迹与修正轨迹
可以看出,车辆行驶轨迹与未经数据预处理时比

较有了明显的改善,这说明文中的地图匹配算法对车辆行驶数据进行预处理可以取得较好的效果。

文中用匹配时间、匹配率以及匹配完成后数据点剩余比例三个指标来衡量算法的性能^[16]。匹配时间指完成数据预处理以及数据点匹配所用总时间;匹配率指数据点匹配到真实行驶道路上的比例;数据点剩余比例指匹配完成后未被当作异常点剔除的数据点占总数据点的比例。

将文中算法、不使用 geohash 函数与异常点填充的确定性匹配算法以及依据概率统计的不确定性匹配算法分别在 10 份行驶数据上运行,运行的硬件条件是具有 Inter Core i7-7700HQ CUP 的个人计算机。实验结果如图 5 ~ 图 7 所示。

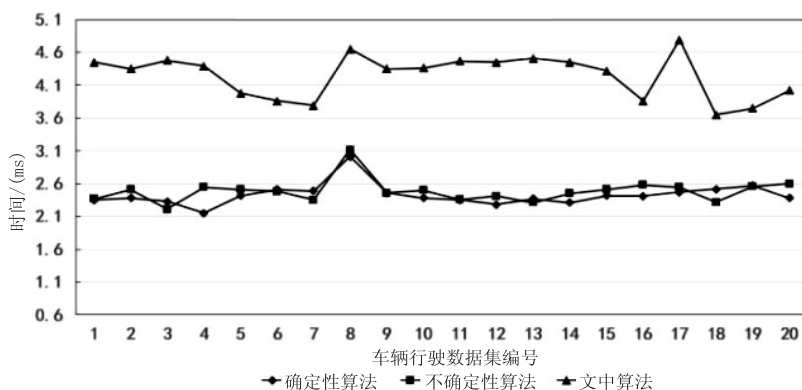


图5 三种算法匹配时间对比

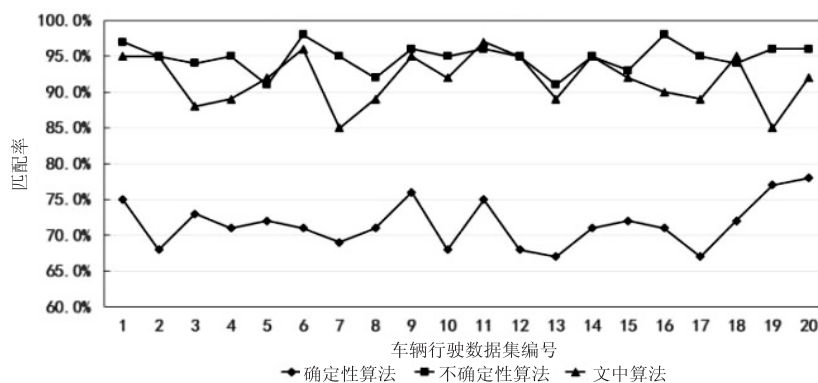


图6 三种算法匹配率对比

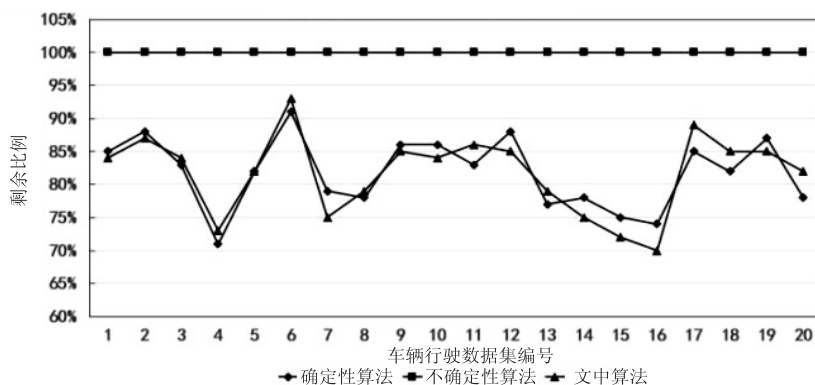


图7 三种算法剩余比例对比

图 5 表示三种算法所使用的匹配时间。可以看出,文中算法所用匹配时间虽然比普通确定性匹配算法略长,但远小于依概率统计的不确定性匹配算法。

图 6 表示三种算法的匹配率。可以看出,文中算法的匹配率稍高于依概率统计的不确定性匹配算法,远高于普通确定性匹配算法。

图 7 表示三种算法的数据点剩余比例。可以看出,由于进行了异常数据点填充,文中算法的剩余比例永远是 100%,远大于其余两种算法,原因是文中算法对异常数据点进行了按比例填充,将剔除的数据点全部填充回原始数据集。

综上所述,文中使用的基于哈希和路网边权的地图匹配算法综合了效率和准确率两个优点,并且保留了大部分的原始数据。可以满足对车辆行驶数据的预处理,同时也可以推广至车辆的实时定位修正。

3 结束语

依据地图匹配算法对车辆行驶数据进行修正对分析车辆驾驶行为起着至关重要的作用。文中设计的新型地图匹配算法,利用 geohash 函数筛选数据点附近的道路,并且添加滑动窗口来减小匹配错误率,最后通过对路网边权进行修正来填充缺失数据点。原理易懂,实现简单,并且具有传统确定性地图匹配算法不具有的高准确率和高效性,不仅能够很大程度上提高车辆行驶数据的质量,还能应用于车辆行驶过程中的实时定位修正,有很高的可推广性。

目前该地图匹配算法中两处用到了加权法,一处是计算数据点与道路的匹配度,另一处是依据“highway”属性来修改邻接表。但权重的确定对加权法的使用有很大的要求,文中权重的确定主要依靠人为经验,这会导致算法缺乏理论依据,如何确定最佳权重来提高算法的匹配率,将会是未来研究的重点。

参考文献:

- [1] 孙 川,吴超仲,褚端峰,等. 基于车联网数据挖掘的运营车辆驾驶速度行为聚类研究[J]. 交通运输系统工程与信息,2015,15(6):82-87.
- [2] 文江辉,杨先敏,吴超仲. 大数据环境下运营车辆驾驶行为

特征分析[J]. 交通信息与安全,2018,36(4):42-50.

- [3] 刘兴权,金美含. 地图匹配算法综述[J]. 科技信息,2014(4):64-65.
- [4] 谢博晖,吴健平. GPS 轨迹数据纠偏方法研究[J]. 计算机技术与发展,2012,22(7):223-226.
- [5] HAN Haihua, MEI Wenbo. A high efficient map-matching algorithm for the GPS data processing intended for the highways[J]. International Journal of High Performance Computing and Networking, 2019, 13(2):132-140.
- [6] 宋 洁,李国燕,李娜娜,等. 基于模糊逻辑的 GPS/DR 地图匹配算法[J]. 计算机工程与科学,2008,30(10):30-32.
- [7] 滕志军,曲兆强,侯学艳,等. 基于曲线拟合与拓扑结构的地图匹配算法[J]. 计算机工程,2018,44(8):291-295.
- [8] TAGUCHI S, KOIDE S, YOSHIMURA T. Online map matching with route prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(1):338-347.
- [9] YANG C, GIDÓFALVI G. Fast map matching, an algorithm integrating hidden Markov model with precomputation[J]. International Journal of Geographical Information Science, 2018, 32(3):547-570.
- [10] 向隆刚,王德浩,龚健雅. 大规模轨迹数据的 Geohash 编码组织及高效范围查询[J]. 武汉大学学报:信息科学版, 2017, 42(1):21-27.
- [11] SHARATH M N, VELAGA N R, QUDDUS M A. A dynamic two-dimensional (D2D) weight-based map-matching algorithm[J]. Transportation Research Part C: Emerging Technologies, 2019, 98:409-432.
- [12] 陈 滨. 基于融合函数构建的 GPS 车辆轨迹地图匹配算法[J]. 许昌学院学报, 2018, 37(12):1-4.
- [13] 丁涛杰,史殿习,李永谋. 基于出租车 GPS 数据的道路平均速度估计方法[J]. 计算机技术与发展, 2015, 25(7):15-19.
- [14] 张校慧,孙 凯,职保平,等. 针对复杂道路网络的车辆轨迹地图匹配算法[J]. 测绘科学, 2018, 43(8):110-115.
- [15] 陈 菲,邹 涛,王 伦. 改进的基于投影的地图匹配算法[J]. 计算机工程与应用, 2011, 47(31):242-244.
- [16] MILLARD-BALL A, WEINBERGER R R. Map-matching poor-quality GPS data in urban environments: the pg-MapMatch package[J]. Transportation Planning and Technology, 2019, 42(6):539-553.