

2-16-2023

Efficient HMM Map Matching Method Using R-tree and Trajectory Segmentation

Yanjiao Song

1.College of Hydrology and Water Resources, Hohai University, Nanjing 210098, China;;
songyanjiao2000@163.com

Jiayue Zhou

1.College of Hydrology and Water Resources, Hohai University, Nanjing 210098, China;

Longhao Wang

1.College of Hydrology and Water Resources, Hohai University, Nanjing 210098, China;

Jing Wu

1.College of Hydrology and Water Resources, Hohai University, Nanjing 210098, China;

See next page for additional authors

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research](#), [Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Efficient HMM Map Matching Method Using R-tree and Trajectory Segmentation

Abstract

Abstract: In view of the incapability of traditional methods to efficiently process massive trajectory data, an improved *HMM (hidden-Markov model)* map matching algorithm is proposed. Spatial index for road networks is established through *R-tree spatial index*. *GPS trajectory data are segmented* based on the position change rates of trajectory points. R-tree index is used to quickly determine the candidate road section that sub-trajectories belong to, and the *key points of the sub-trajectories* instead of the entire sub-trajectories are selected to judge which road the sub-trajectories should be matched with. The map matching of each sub-trajectory is carried out on the basis of the former results. The algorithm is verified by a simulation experiment using Beijing's floating car data and OpenStreetMap data. Experimental result shows that the proposed algorithm can reduce the workload of road search and trajectory point traversal and can greatly improve the algorithmic efficiency.

Keywords

hidden-Markov model, map matching, R-tree, trajectory segmentation, GPS trajectory data, road networks

Authors

Yanjiao Song, Jiayue Zhou, Longhao Wang, Jing Wu, Rui Li, and Xiaoping Rui

Recommended Citation

Yanjiao Song, Jiayue Zhou, Longhao Wang, Jing Wu, Rui Li, Xiaoping Rui. Efficient HMM Map Matching Method Using R-tree and Trajectory Segmentation[J]. Journal of System Simulation, 2023, 35(2): 339-349.

采用 R 树和轨迹分段的 HMM 高效地图匹配方法

宋纘蛟¹, 周佳悦¹, 王龙浩¹, 吴婧¹, 李睿¹, 芮小平^{2*}

(1. 河海大学 水文水资源学院, 江苏 南京 210098; 2. 河海大学 地球科学与工程学院, 江苏 南京 211100)

摘要: 针对传统隐马尔可夫模型(hidden-Markov model, HMM)地图匹配算法无法高效处理大量轨迹数据的问题, 提出了一种改进的HMM地图匹配算法。采用R树空间索引方法为路网建立空间索引, 基于轨迹点位置变化率对GPS轨迹数据进行分段, 并利用R树索引快速确定子轨迹所属的候选路段, 在子轨迹中挑选关键点代替整段子轨迹判断所属路段, 根据结果完成各子轨迹的地图匹配。仿真结果表明: 与传统HMM地图匹配算法相比, 改进算法可以同时减少道路搜索和轨迹点遍历的工作量, 大幅提高算法效率。

关键词: 隐马尔可夫模型; 地图匹配; R树; 轨迹分段; GPS轨迹数据; 道路网络

中图分类号: TP310.6

文献标志码: A

文章编号: 1004-731X(2023)02-0339-11

DOI: 10.16182/j.issn1004731x.joss.21-1020

引用格式: 宋纘蛟, 周佳悦, 王龙浩, 等. 采用R树和轨迹分段的HMM高效地图匹配方法[J]. 系统仿真学报, 2023, 35(2): 339-349.

Reference format: Song Yanjiao, Zhou Jiayue, Wang Longhao, et al. Efficient HMM Map Matching Method Using R-tree and Trajectory Segmentation[J]. Journal of System Simulation, 2023, 35(2): 339-349.

Efficient HMM Map Matching Method Using R-tree and Trajectory Segmentation

Song Yanjiao¹, Zhou Jiayue¹, Wang Longhao¹, Wu Jing¹, Li Rui¹, Rui Xiaoping^{2*}

(1. College of Hydrology and Water Resources, Hohai University, Nanjing 210098, China;

2. School of Earth Sciences and Engineering, Hohai University, Nanjing 211100, China)

Abstract: In view of the incapability of traditional methods to efficiently process massive trajectory data, an improved *HMM (hidden-Markov model)* map matching algorithm is proposed. Spatial index for road networks is established through *R-tree spatial index*. *GPS trajectory data are segmented* based on the position change rates of trajectory points. *R-tree index* is used to quickly determine the candidate road section that sub-trajectories belong to, and the *key points of the sub-trajectories* instead of the entire sub-trajectories are selected to judge which road the sub-trajectories should be matched with. The map matching of each sub-trajectory is carried out on the basis of the former results. The algorithm is verified by a simulation experiment using Beijing's floating car data and OpenStreetMap data. Experimental result shows that the proposed algorithm can reduce the workload of road search and trajectory point traversal and can greatly improve the algorithmic efficiency.

Keywords: hidden-Markov model; map matching; R-tree; trajectory segmentation; GPS trajectory data; road networks

收稿日期: 2021-10-04 修回日期: 2021-12-21

基金项目: 国家自然科学基金(41771478)

第一作者: 宋纘蛟(2000-), 男, 硕士生, 研究方向为地理大数据处理与分析。E-mail: songyanjiao2000@163.com

通讯作者: 芮小平(1975-), 男, 教授, 博士, 博士生导师, 研究方向为地理信息系统理论与应用。E-mail: ruixpsz@163.com

0 引言

GPS 轨迹数据是带有时间和空间信息的序列数据, 数据获取成本低, 覆盖范围广, 且拥有时态特性。通过 GPS 轨迹数据既可以进行微观个体活动模式的研究, 也可以进行宏观活动系统的城市空间结构研究, 已成为城市地理大数据研究的重要数据源^[1-4]。由于 GPS 数据的误差, 未校准的轨迹数据往往会偏离实际道路, 在实际应用时, 要将 GPS 数据进行地图匹配, 获得结合路网的轨迹描述^[5]。GPS 轨迹数据的地图匹配是学者对其进行进一步挖掘分析的基础工作之一^[6-9]。

传统的地图匹配算法仅根据几何与拓扑关系将轨迹点匹配到路网上^[10-11], 这种方法仅考虑 GPS 点与道路的空间位置, 而未考虑道路的连通性、双向性等信息, 因此会导致结果产生较大的误差。Newson 等^[12]提出了基于隐马尔可夫模型(hidden Markov model, HMM)的地图匹配算法, 该算法考虑到了车辆行驶的历史信息, 从概率论的角度引入了观测概率和隐藏概率, 很好地解决了单纯几何匹配精度不够的问题, 已经成为 GPS 轨迹地图匹配的主要方法。目前, 众多地图匹配算法的工作是在 HMM 地图匹配算法的基础上对其精度和效率进行改进。

对 HMM 地图匹配算法进行精度改进的方法有多种形式, 主要是通过优化维特比算法中的概率计算实现准确率的提高, 如陈浩等^[13]使用动态规划的方法, 将计算单个轨迹点的发射概率和转移概率改进为计算路段联合概率。传统的 HMM 地图匹配算法并未考虑行驶方向对模型的影响, 而刘旻^[14-15]等都提出了将行驶方向(航向角)引入转移概率计算的方法。另有学者对匹配前或匹配后的轨迹数据进行一定处理, 如 Zhou 等^[16]考虑到实际交通状况较为复杂, 提出在匹配前选出匹配易错点并进行人工注释, 其余点正常进行 HMM 地图匹配。吴刚等^[17]使用遗传算法对 HMM 地图匹配的结果进行处理, 进一步提高了匹配路段的准确性。

实际应用场景中, 尤其是在线地图应用中, 对轨迹数据地图匹配的时效性要求很高, 因此, 许多学者更加侧重于研究如何提高 HMM 地图匹配的效率, 如 Huang 等^[18]提出了从历史轨迹数据中识别出频繁模式的方法, 刘旻^[14, 19]等都选择对轨迹数据进行分割处理以减少转移概率的计算次数。也有学者对道路搜索的速率来进行优化, 如王晓蒙等^[15]对路网数据建立了哈希索引, 可以在一定程度上提高地图匹配的效率。另一种解决办法是对轨迹点进行抽稀处理, 即对轨迹点进行抽样, 比如每隔 50 m 保留一个轨迹点, 只对抽稀后的样本点进行匹配。这种方法可以明显提高匹配速度, 但缺点是没有考虑到轨迹点的各个属性(速度、方向等)与所在路段的相关性。

综上所述, 众多学者针对地图匹配算法的高效率要求对 HMM 地图匹配算法进行了改进, 已有的方法或对 GPS 轨迹数据的遍历进行简化处理以减少对轨迹点的循环次数, 或对路网数据建立索引以加快搜索速率。实际上, 地图匹配算法的时间复杂度主要源于对路网数据的搜索和对轨迹数据的逐点判断, 若从这两方面综合起来进行改进, 可以极大地提高算法的效率。因此, 本文设计一种基于 R 树和轨迹匹配的 HMM 地图匹配方法, 对路网数据建立 R 树索引的同时, 也对 GPS 轨迹数据进行了轨迹分割处理, 实现从两方面同时提高算法的效率。

1 HMM 为核心的地图匹配

HMM 由马尔可夫链演变而来。在研究随机变量的时间序列 $X_1, X_2, \dots, X_n$ (即随机过程)时, 马尔可夫为了简化问题提出了一种假设, 即随机过程中各个状态 X_n 的概率分布只与它的前一个状态 X_{n-1} 有关, 即

$$P(X_n | X_1, X_2, \dots, X_{n-1}) = P(X_n | X_{n-1}) \quad (1)$$

这个假设被命名为马尔可夫假设, 而符合这个假设的随机过程被称作马尔可夫过程, 也称为马尔可夫链。鲍姆等在此基础上提出了 HMM, 即

任意时刻 n 的状态 X_n 是不可见的。在HMM中, 观察者没法通过观察到一个状态序列 $X_1, X_2, \dots, X_n$ 来推测转移概率等模型参数, 但在每个时刻 n 该模型会输出一个符号 Y_n , 该符号仅和 X_n 相关, 如图1所示。模型中, $X_1, X_2, \dots, X_n$ 为隐藏变量, 其参数不能直接获取, 共同组成隐藏序列; 而 $Y_1, Y_2, \dots, Y_n$ 为观测变量, 其参数可见, 共同组成观测序列; 从前一个状态进入当前状态的概率 $P(X_n|X_{n-1})$ 称为转移概率; 每个状态产生相应输出符号的概率 $P(Y_n|X_n)^{[20]}$ 。HMM的一个典型用法就是给定一个模型和某个特定的观测序列, 求出最可能产生这个观测序列的隐藏序列。

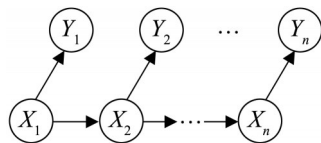


图1 HMM
Fig. 1 HMM

在地图匹配场景中, GPS轨迹点的位置可以直接获得, 因而作为观测序列; 待匹配路段是未知的, 因而作为隐藏序列, 2个序列共同组成一个隐马尔可夫模型。每个GPS轨迹点是固定的, 但有多条路段供其匹配。对应到HMM中, 每个状态输出对应的观测变量 Y_n 是固定的(即GPS轨迹点), 但是这个状态的值可能有多个路段可供选择。用 X_{ij} 表示状态 X_i 的第 j 个可能的值, 把每个状态按照不同的值展开, 就得到一个篱笆网络(Lattice), 如图2所示。图中, 每个状态有3个或4个值, 表示每个GPS轨迹点有3个或4个可能的路段与之匹配(实际中可以有任意个)。不同状态间的转移用路径(Path)表示, 从第一个状态到最后一个状态的任何一条路径都可能产生观测路径 Y , 而每条路径的可能性不一样, 求出最可能的隐藏序列就是要找到最可能的一条路径。求出每条路径的可能性并不困难, 但由于路径组合数非常庞大,

比较各个路径可能性的计算量会呈指数爆炸式增长。著名的维特比算法就是一个在求解HMM时能使运算量和状态数目成正比的算法。

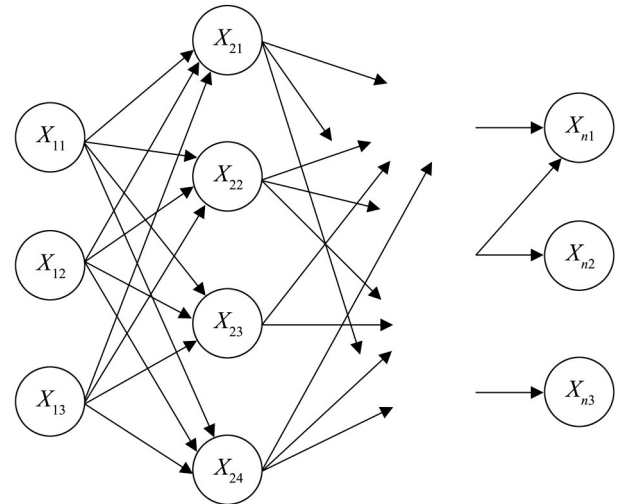


图2 篱笆网络
Fig. 2 Lattice

维特比算法的步骤如下(j, k, l 用来代表各个状态的某个节点):

- (1) 从观测序列的起点 S 出发(在地图匹配中为第一个待匹配的轨迹点), 对于第1个状态的所有节点 X_{1j} , 计算出 S 到它们的距离 $d(S, X_{1j})$ 。
- (2) 对于第2个状态的所有节点 X_{2k} , 计算从 S 到它们的最短距离。对应的每条路径长度为 $d(S, X_{2k}) = d(S, X_{1j}) + d(X_{1j}, X_{2k})$, 对于各个 X_{1j}, X_{2k} 状态的路径长度进行计算, 找到最小值。
- (3) 按照与(2)类似的方法走到第3个状态 X_3 , 直至最后一个状态 X_n , 得到距离最短的 X_{nl} 。由此进行回溯, 记录通往 X_{nl} 的每个节点, 就得到了从头到尾的最短路径, 也就是最可能的隐藏序列。

算法中的“距离”指从一个匹配路段转到另一个的可能性, 可以由各个状态观测到各个节点的概率 $P(X_{ij})$ 的累积进行度量。根据马尔可夫假设, 当前状态的概率分布只与前一个状态有关, 即为前一个状态的概率乘以2个状态间的转移概率, 即

$$P(X_i) = P(X_{i-1}) \cdot P(X_i|X_{i-1}) \quad (2)$$

要求解状态内部的某个节点的概率, 需要在

篱笆网络中考虑, 将当前状态的概率乘以在当前状态观测到该节点的概率, 即

$$P(X_{ij}) = P(X_{i-1}) \cdot P(X_i | X_{i-1}) \cdot P(X_{ij} | X_i) \quad (3)$$

将维特比算法应用到地图匹配情景中, $P(X_{ij})$ 为选择当前路段的概率; $P(X_{i-1})$ 为上一路段的概率; $P(X_i | X_{i-1})$ 为两状态间的转移概率, 用来度量从上一个点到这一个点的前进过程中, 从上一个点对应的某个路段到这个点对应的某个路段的代价, 可由两轨迹点之间的距离和两个路段上对应点之间距离之差度量; $P(X_{ij} | X_i)$ 为当前状态中选择当前路段的概率, 轨迹点离路段越远、前进方向与路段方向相差越小, 该概率越大。

2 基于轨迹匹配的 HMM 地图匹配算法

传统的 HMM 地图匹配算法比简单的距离匹配具有更高的效率和准确性。然而, 在实际应用中, GPS 轨迹数据的数量级可达百万、千万甚至更高, 路网数据同样错综复杂, 传统 HMM 地图匹配算法对于所有状态节点进行遍历并计算观测概率的机制并不能满足对于效率的要求。因此, 本文分别应用 R 树索引和轨迹分段对该算法进行了改进。

2.1 针对路网数据构造 R 树索引

为了提高空间数据的搜索效率, Guttman^[21] 提出了 R 树索引方法(R-trees)。R 树是一种采用空间分割概念的平衡树, 通过最小包围矩形(MBR)将多维空间分割为多个层次, 可以大幅减少空间搜索中结点的访问量。在构建一棵 R 树时, 每个叶子结点保存指向空间中各地物 MBR 的指针; 倒数第 2 层同样保存指向 MBR 的指针, 这类 MBR 包含一个或多个各地物的 MBR; 以此类推, 最终在整个空间建立索引。在基于 HMM 的地图匹配算法中, 每个状态都可能大量的观测结果, 即可能的匹配路段。匹配过程中, 遍历每个 GPS 轨迹点时, 首先要做的就是筛选出候选匹配路段, 即

当前状态的所有节点。通常情况下, 路网复杂、路段繁多, 因而, 此搜索步骤会耗费大量时间。若在匹配前以各个路段为最小地物单元建立 R 树索引, 则可大大缩短候选路段的搜索时间, 提高算法效率。

图 3 为针对示例路网的分割方法。其中红色为最上层空间, 被分为 R1, R2; 蓝色为第 2 层空间; 黑色为第 3 层, 即各个路段的 MBR。该 R 树的数据结构示意图如图 4 所示。

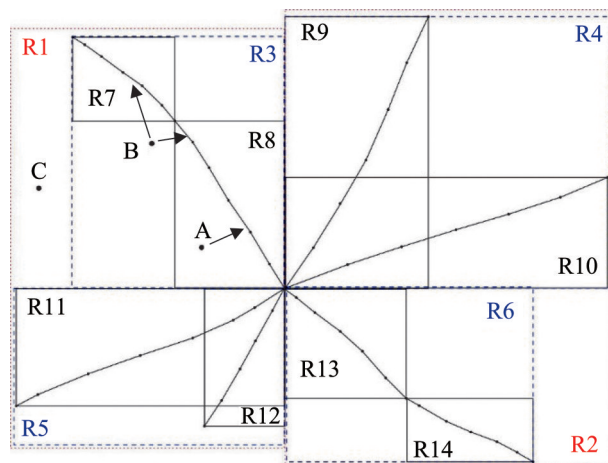


图3 针对路网的空间分割

Fig. 3 Spatial division for road network

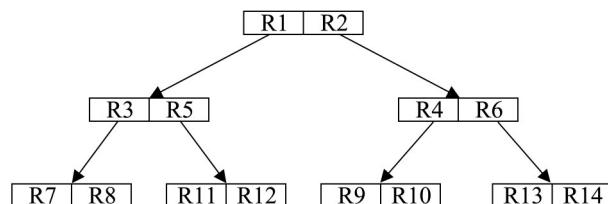


图4 为图3建立的 R 树

Fig. 4 R-tree built for example in Fig. 3

在地图匹配过程中, 可以通过 GPS 轨迹点的地理坐标快速找到其所在的大矩形(即处于非叶子结点的 MBR), 继而根据 R 树存储的指针找到该矩形内所有路段, 将它们作为候选路段。图 3 中, A, B, C 为待匹配的 3 个数据点, 体现了搜索待匹配路段时的 3 种情况。容易看出, A 点位于 R8 矩形即叶子结点的 MBR 内, 则可根据 R 树寻找上一层的矩形即 R3, 继而将 R3 内的 2 个路段列为当

前状态的候选路段; B点位于R3内, 则直接把R7、R8的对应路段列为候选; C点处于R1中, 则将其中的R7、R8、R11、R12对应的路段都列为候选路段。逐点通过R树搜索候选路段在一定程度上提高了效率, 但仍然存在轨迹点数量大、遍历时间长的问题, 因此, 本文进一步提出了基于R树和子轨迹的快速HMM地图匹配方法。

2.2 基于R树和子轨迹的快速HMM地图匹配实现方法

轨迹点的数据来源为GPS接收器的连续采集, 在同一路段上, 相邻轨迹点在行驶方向、行驶速度等方面都具有较强的相似性; 在不同路段上, 相邻轨迹点的各个属性则会有较明显的变化。如果将出行轨迹中性质相似的相邻轨迹点组成子轨迹, 在各个子轨迹内部进行抽稀处理, 此样本便可以极大程度地代替子轨迹。对抽稀产生的样本进行候选路段的搜索, 并将其作为整个子轨迹的候选路段, 便可以大幅提高搜索效率。同时, 相比于简单的抽稀处理, 这种方法能更准确地代表整个轨迹。

子轨迹的划分通常由比较相邻点轨迹点的相似性实现, 相似性超过规定阈值的点视为子轨迹点。相似性的计算很大程度上会受到轨迹点定位误差的影响, 因而子轨迹的划分算法也应具有数据清洗功能。本文根据轨迹数据的位置实时变化率对GPS数据进行逐点处理, 通过点间位置变化率与平均变化率的大小关系判断该点是否处于当前子轨迹。两点间的位置变化率又可分为经度平均变化率和纬度平均变化率。二者计算方法均为该点的经(纬)度值减去上一点的经(纬)度值的绝对值, 再除以上一点的经(纬)度值:

$$r(i) = \frac{|L(i) - L(i-1)|}{L(i-1)}, 2 < i < n \quad (4)$$

具体算法如下:

step 1: 创建一个子轨迹并作为当前子轨迹, 从第1个点或当前点开始将前2个轨迹点加入, 第

3个轨迹点设为当前点。

step 2: 从当前点*i*开始遍历GPS轨迹点。

step 3: 计算当前点的位置变化率和当前子轨迹的平均位置变化率。平均位置变化率为

$$R(n) = \frac{1}{n-1} \sum_{i=2}^n r(i) \quad (5)$$

step 4: 判断当前点的位置变化率和当前子轨迹的平均位置变化率的大小关系, 如图5所示。当*i*点满足 $r(i)_{\text{lon}} \leq 1.5R(n)_{\text{lon}}$ 且 $r(i)_{\text{lat}} \leq 1.5R(n)_{\text{lat}}$ 时, 说明*i*点位于“1区”, 位置变化率与当前子轨迹的平均值相差不大, 认为其属于当前子轨迹, 转到step 5; 否则, 结束当前子轨迹, 返回step 1。

step 5: 将当前点加入当前轨迹, 下一点作为当前点, 返回step 2。

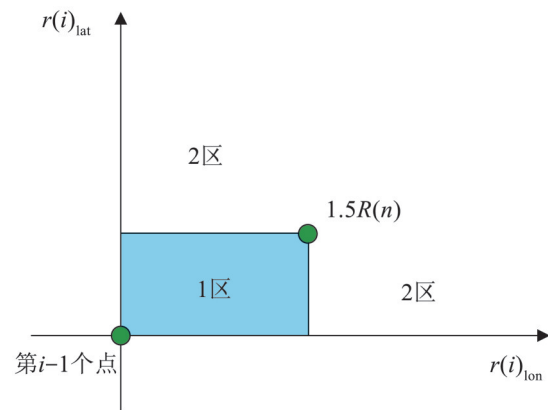


图5 位置变化率分区
Fig. 5 Zone of position change rate

传统HMM地图匹配必须遍历所有轨迹点, 而本算法只要事先划分好子轨迹, 匹配时抽取此子轨迹的样本并对其搜索候选道路, 便可得知全部子轨迹点的候选道路, 这样可以大大提高算法的效率。由于子轨迹划分时考虑到了车辆行驶状态, 而车辆行驶状态很大程度上受到路段特性的影响, 因而可以近似认为每个子轨迹同属于一个路段。具体来讲, 在子轨迹内部, 两端的轨迹点由于处在2个子轨迹的交界处, 不一定同属于一个路段; 但可以认为子轨迹中部的轨迹点可以匹配到同一路段中。因此, 在抽样时, 可以在子轨

迹两端抽取较多样本,而在子轨迹中部抽取尽量少的样本,以实现效率的最大化提高。

完整的基于 R 树和子轨迹的 HMM 地图匹配算法步骤如下:

step 1: 对于路网数据建立 R 树。

step 2: 对 GPS 轨迹数据进行子轨迹的划分。

step 3: 遍历所有子轨迹,对于每个子轨迹,将所有轨迹点等分为 4 部分,第 1 部分和第 4 部分保留所有点(即保留子轨迹的首尾),而在中间 2 个部分每 2 个点取样一个点,按顺序排列,生成整个轨迹的样本。

step 4: 使用维特比算法计算该样本的最优匹配道路:

(1) 从样本起点出发,利用 R 树索引寻找候选路段;综合考虑轨迹点与各路段距离、前进方向的差异性,通过公式计算选择各个路段的概率:

$$P(X_{ij}) = \frac{1/(d_j \cdot d_\theta)}{\sum 1/(d_j \cdot d_\theta)} \quad (6)$$

式中: d_j 为与第 j 个路段间的距离,取轨迹点到道路的垂线长度; θ_j 为轨迹点与第 j 个路段间的前进方向夹角,取正值。

(2) 从第 2 个样本点开始,通过式(3)计算各个候选路段的观测概率 $P(X_{ij})$,直至最后一个样本点。

(3) 找到最后一个状态中具有最大观测概率的候选路段,并由此进行回溯,得到各个状态的相应候选路段,直到样本起点。

step 5: 将回溯得到的整条路径作为该子轨迹的匹配路径,并将子轨迹内的各个点匹配到该路径的距离最小点上,完成匹配。

图 6 为上述地图匹配算法的示意图。a-b-c-d 为一段连续轨迹(为方便展示,此处只绘出少数轨迹点),R13、R14 中分别有 2 个路段,共同被 R6 包围。根据本文的子轨迹划分规则,得到 a、b、c、d 4 段子轨迹。对于 a、b、d,可以按照上文的方法,寻找非叶子结点矩形即 R6,将其中的路段

列为候选;对于 c,由于其不在任一叶子结点内,直接将 R6 中的矩形列为候选;之后利用维特比算法计算各个候选概率,找到最大概率的路段,回溯得到整个路段;最后将轨迹内各个点匹配到相应路段的距离最小点上。

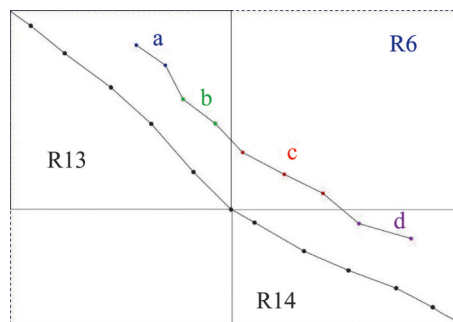


图 6 GPS 轨迹数据分段匹配

Fig. 6 GPS trajectory segment matching

3 仿真实验与结果分析

3.1 数据源

本文使用的路网数据为北京市的 OpenStreetMap 数据;轨迹数据为微软 2012 年发布的北京市 GPS 轨迹数据集。

3.2 软硬件环境

为了验证本算法的适用性,本次仿真选择普通配置的计算机进行实验,硬件环境方面,电脑型号为 Legion Y7000P 2019, CPU 为 Intel(R) Core (TM) i7-9750H CPU @ 2.60 GHz 2.59 GHz,内存为 16.0 GB。软件环境方面,编程语言 Python 的版本为 3.8,电子地图采用 ArcGIS 软件显示。

3.3 仿真实验

3.3.1 建立 R 树

按照 2.1 中的方法对 OpenStreetMap 数据建立 R 树索引。经过统计,共有 1 827 个路段,因而创建的 R 树有 1 827 个叶子结点。

3.3.2 划分子轨迹

按照 2.2 中的方法进行子轨迹的划分。以一条

由 932 个 GPS 点组成的轨迹为例, 按照式(4)进行经纬度变化率的求解, 部分结果如表 1 所示。经纬度变化率从第 2 个轨迹点开始计算, 故轨迹首点(表格第 1 行)经纬度变化率为空。

表 1 经纬度变化率
Table 1 Latitude and longitude change rate

序号	北纬/(°)	东经/(°)	纬度变化率	经度变化率
1	39.978 728	116.344 453		
2	39.978 682	116.344 456	-1.151×10^{-6}	2.579×10^{-8}
3	39.978 617	116.344 469	-1.626×10^{-6}	1.117×10^{-7}
4	39.978 685	116.344 443	1.701×10^{-6}	-2.235×10^{-7}
5	39.978 913	116.344 425	5.703×10^{-6}	-1.547×10^{-7}
6	39.979 310	116.344 444	9.930×10^{-6}	1.633×10^{-7}
7	39.979 486	116.344 405	4.402×10^{-6}	-3.352×10^{-7}
8	39.979 509	116.344 393	5.753×10^{-7}	-1.031×10^{-7}
9	39.979 598	116.344 397	2.226×10^{-6}	3.438×10^{-8}
10	39.979 493	116.344 384	-2.626×10^{-6}	-1.117×10^{-7}

按照式(5)求解该轨迹的经纬度平均变化率, 纬度平均变化率为 1.60×10^{-6} , 经度平均变化率为 6.78×10^{-7} 。比较经纬度变化率与经纬度平均变化率的大小, 便可判断各个点是否为子轨迹起点。分析结果如表 2 所示。轨迹首点无经纬度变化率, 故表格第一行的经纬度判断为空值, 并将轨迹首点作为非起点处理。

表 2 子轨迹起点的判断
Table 2 Judgement of sub-trajectory starting point

序号	纬度判断	经度判断	子轨迹起点
1			否
2	<	<	否
3	<	<	否
4	<	<	否
5	>	<	是
6	>	<	是
7	>	<	是
8	<	<	否
9	<	<	否
10	>	<	是

注: “<” 为轨迹点经纬度变化率小于平均变化率; “>” 为轨迹点经纬度变化率大于平均变化率。

处理后, 整个轨迹的 932 个点被划分为 26 个

子轨迹, 最长子轨迹涵盖 106 个轨迹点, 最短子轨迹只有 3 个轨迹点。

3.3.3 候选路段搜索

用其中 1 个子轨迹进行仿真实验, 其轨迹点坐标和行驶方向如表 3 所示, 其中行驶方向的角度从正东按逆时针方向度量, 90° 为正北。首先按照算法步骤对该子轨迹进行抽样处理, 得到表 4 所示的样本。后续实验通过该样本进行展示。

表 3 某一子轨迹
Table 3 A certain sub-trajectory (°)

序号	北纬	东经	行驶方向
1	39.981 675	116.336 576	93
2	39.981 715	116.336 470	89
3	39.981 727	116.336 485	92
4	39.981 727	116.336 486	95
5	39.981 726	116.336 483	90
6	39.981 727	116.336 501	97
7	39.981 728	116.336 512	91
8	39.981 729	116.336 525	86
9	39.981 729	116.336 523	85
10	39.981 730	116.336 528	87

表 4 子轨迹的样本
Table 4 Sample of sub-trajectory (°)

序号	北纬	东经	行驶方向
1	39.981 675	116.336 576	93
2	39.981 715	116.336 470	89
4	39.981 727	116.336 485	95
6	39.981 727	116.336 501	97
8	39.981 729	116.336 525	86
9	39.981 729	116.336 525	85
10	39.981 730	116.336 528	87

图 7 展示了一段轨迹, 其中蓝色矩形为子轨迹的外包矩形, 包围的 10 个点为该子轨迹, 矩形中红色点为表 4 所示的抽样结果。红色矩形为 R 树叶子结点, 橙色矩形为叶子结点上一级的父结点。使用 R 树搜索该子轨迹的候选路段, 发现子轨迹外包矩形与路段 1、4、5 的外包矩形相交; 进一步搜索它们的父结点, 发现父结点的橙色矩形也包括了路段 2、3、6 的外包矩形。因此将路段 1~6

作为该子轨迹的候选道路。如此确定整个子轨迹的候选路段，后续便无需逐子轨迹点搜索候选路段，大幅提高了算法效率。

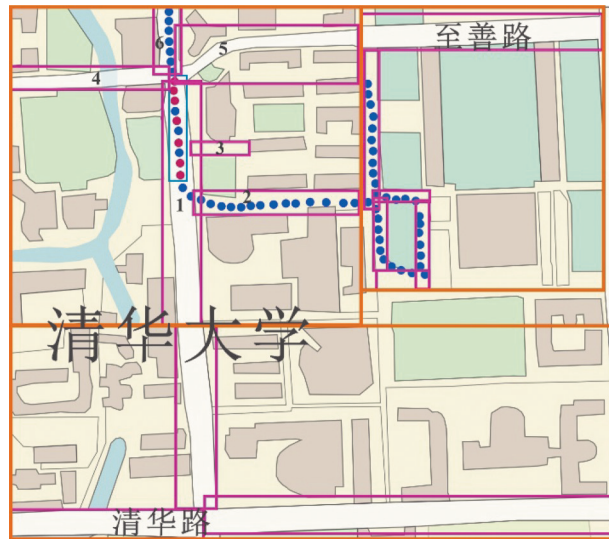


图7 候选路段
Fig. 7 Candidate roads

3.3.4 候选路段概率计算

通过路段数据点可计算出路段方向，如表5所示。首先使用式(6)计算样本首点的6个状态概率，如选择候选路段1时， θ_j 为 2° ， d_j 为1.8 m，计算得到该节点概率为0.831 2；之后计算第2个状态的概率，根据式(3)得到共36个节点的概率；后续点同理，直到最后一点，最后一个状态的节点概率如表6所示，由于节点过多，只做部分展示。表中路径记录了到达该节点所经过的各个节点，概率则为通过该路径到达末尾状态的概率。从其中选择最大值，并进行回溯，发现各个状态选择的候选路段都为路段1。连接各个状态的候选路径得到的仍为路段1，最后子轨迹就匹配到路段1上。

表5 候选路段方向
Table 5 Direction of candidate roads

候选路段	方向/(°)	候选路段	方向/(°)
1	95	4	173
2	7	5	41
3	6	6	89

表6 末尾状态概率计算

Table 6 End state probability calculation

节点编号	概率	路径
1	0.443 2	1-1-1-1-1-1-1
2	0.048 7	1-3-6-1-1-1-1
3	0.023 6	1-5-2-1-6-1-1
4	0.001 5	2-1-5-4-6-3-1
5	0.000 1	5-3-2-4-5-1-6

3.4 仿真结果与分析

图8展示了改进算法于北京两处取样路段的匹配效果，其中蓝色点为匹配后的轨迹点。



图8 部分轨迹点的匹配效果
Fig. 8 Matching result of some trajectory points

由图8可见, 即使在比较复杂的路网, 如校园内, 轨迹点也基本全部匹配到了道路上, 说明改进算法的精度可以满足实际应用的要求。本文提出的改进算法致力于提高地图匹配效率, 因此不对精度做进一步讨论, 而在下文着重进行效率分析。

为验证算法效率, 从实际GPS轨迹数据中截

取1 000、5 000、10 000、50 000、100 000、500 000、1 000 000个连续轨迹点进行实验, 分别使用传统的HMM地图匹配算法、使用R树改进的HMM地图匹配算法与使用R树和轨迹分段的HMM地图匹配算法进行仿真实验, 分析结果如图9所示, 其中柱形图代表不同数据量的整体匹配效率, 折线图代表单点匹配效率。

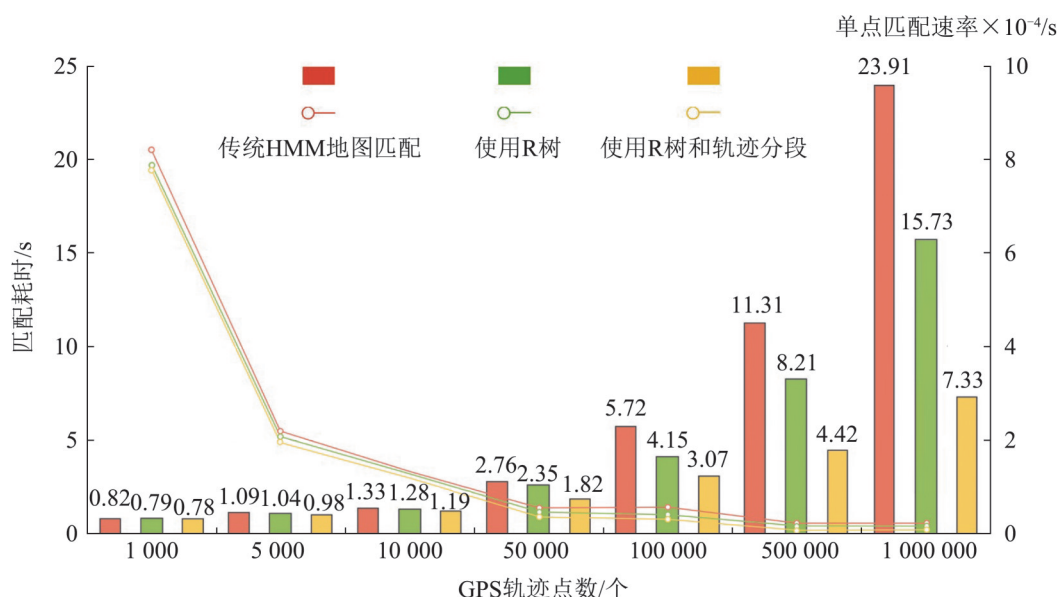


图9 3种匹配算法的效率分析
Fig. 9 Efficiency analysis of 3 matching algorithms

由图9可知, 随着数据量增大单点匹配率逐渐减小, 其主要原因是程序的编译时间在数据量小时占总运行时间的比重很大, 而这种影响在数据量极大时会平均到各个点, 几乎可以忽略不计。数据量较小时3种算法效率相差不大, 但当数据量超过10 000, 仅使用R树对算法效率有一定的提高, 而使用R树和轨迹分段的HMM地图匹配算法具有十分明显的优势; 在数据量为50 000、100 000、500 000、1 000 000时, 使用2种改进的算法运行时间比传统HMM地图匹配算法分别减少了34%、46%、61%和69%。由此可以看出, 本文针对HMM的2种改进都能显著提高地图匹配效率, 并且随着数据量的增大, 这种效率的优势越来越明显。

当下用于智能交通领域的数据具有百万、千万甚至更高的数量级, 而本文提出的改进的HMM地图匹配算法在这种大量轨迹数据的应用场景下更加高效实用。

4 结论

GPS轨迹数据是智能交通领域的重要数据源, 而地图匹配是应用轨迹数据的关键一环。面对海量轨迹数据, 提高地图匹配算法的效率至关重要。本文在传统HMM地图匹配算法的基础上, 结合R树空间索引和轨迹匹配方法, 提出了一种快速地图匹配算法, 同时在路网数据处理和轨迹数据处理两方面进行了效率优化, 使算法在时效性上更

具优势。通过验证,改进算法的效率在GPS轨迹数据量较大的情况下有明显提高,适用于轨迹数据量极大的应用场景。另外,本算法仍有进步空间,比如可以采用并行计算的方法进一步提高地图匹配效率,有待进一步研究。

参考文献:

- [1] Xi Liu, Gong Li, Gong Yongxi, et al. Revealing Travel Patterns and City Structure with Taxi Trip Data[J]. *Journal of Transport Geography*(S0966-6923), 2015, 43 (C): 78-90.
- [2] Yang Zhou, Fang Zhixiang, Thill Jean-Claude, et al. Functionally Critical Locations in an Urban Transportation Network: Identification and Space-Time Analysis Using Taxi Trajectories[J]. *Computers, Environment and Urban Systems*(S0198-9715), 2015, 52: 34-47.
- [3] 朱递,刘瑜.一种路网拓扑约束下的增量型地图匹配算法[J]. *武汉大学学报(信息科学版)*, 2017, 42(1): 77-83.
Zhu Di, Liu Yu. An Incremental Map-Matching Method Based on Road Network Topology[J]. *Geomatics and Information Science of Wuhan University*, 2017, 42(1): 77-83.
- [4] 吴涛,向隆刚,龚健雅.路网更新的轨迹-地图匹配方法[J]. *测绘学报*, 2017, 46(4): 507-515.
Wu Tao, Xiang Longgang, Gong Jianya. Renewal of Roads Networks Using Map-matching Technique of Trajectories[J]. *Acta Geodaetica et Cartographica Sinica*, 2017, 46(4): 507-515.
- [5] Cao Wen, Zhao Jing, Meng Weican, et al. Map Matching Algorithm According to Pseudo-Zernike Moments[C]// *Second World Congress on Software Engineering*. Wuhan: IEEE, 2010: 21-24.
- [6] Zheng Yu. Trajectory Data Mining: An Overview[J]. *ACM Transactions on Intelligent Systems and Technology*(S2157-6904), 2015, 6(3): 29.
- [7] 卢家品,罗月童,黄兆嵩,等.基于排名学习和多源信息的地图匹配方法[J]. *浙江大学学报(理学版)*, 2020, 47 (1): 27-35.
Lu Jiapin, Luo Yuetong, Huang Zhaosong, et al. An Information Fusion Map Matching Method Based on ranking Learning[J]. *Journal of Zhejiang University (Science Edition)*, 2020, 47(1): 27-35.
- [8] Ren Ming, Karimi Hassan A. A Fuzzy Logic Map Matching for Wheelchair Navigation[J]. *GPS Solutions* (S1521-1886), 2012, 16(3): 273-282.
- [9] Hu G, Shao J, Liu F, et al. IF-Matching: Towards Accurate Map-Matching with Information Fusion[J]. *IEEE Transactions on Knowledge and Data Engineering* (S2375-026X), 2017, 29(1): 114-127.
- [10] Yoonsik Bang, Kim Jiyoung, Yu Kiyun. An Improved Map-Matching Technique Based on the Fréchet Distance Approach for Pedestrian Navigation Services[J]. *Sensors* (S1424-8220), 2016, 16(10): 1768.
- [11] Ling Yuan, Li Dan, Hu Song. A Map-Matching Algorithm with Low-Frequency Floating Car Data Based on Matching Path[J]. *EURASIP Journal on Wireless Communications and Networking*(S1687-1499), 2018, 2018(1): 146.
- [12] Paul Newson, Krumm John. Hidden Markov Map Matching Through Noise and Sparseness[C]// *17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems*, ACM-GIS Seattle, Washington: Association for Computing Machinery, 2009: 336-343.
- [13] 陈浩,许长辉,张晓平,等.基于隐马尔科夫模型和动态规划的手机数据移动轨迹匹配[J]. *地理与地理信息科学*, 2019, 35(3): 1-8.
Chen Hao, Xu Changhui, Zhang Xiaoping, et al. Hidden Markov Model and Dynamic Programming Based Map Matching Method for Mobile Trajectories Using Mobile Phone Data[J]. *Geography and Geo-Information Science*, 2019, 35(3): 1-8.
- [14] 刘旻,李梅,徐晓宇,等.一种基于HMM模型改进的地图匹配算法[J]. *北京大学学报(自然科学版)*, 2018, 54 (6): 1235-1241.
Liu Min, Li Mei, Xu Xiaoyu, et al. An Improved Map Matching Algorithm Based on HMM Model[J]. *Acta Scientiarum Naturalium Universitatis Pekinensis*, 2018, 54(6): 1235-1241.
- [15] 王晓蒙,池天河,林晖,等.一种面向海量浮动车数据的地图匹配方法[J]. *地球信息科学学报*, 2015, 17(10): 1143-1151.
Wang Xiaomeng, Chi Tianhe, Lin Hui, et al. A Research of Map-Matching Method for Massive Floating Car Data [J]. *Journal of Geo-Information Science*, 2015, 17(10): 1143-1151.
- [16] Zhou Xibo, Ye Ding, Tan Haoyu, et al. HIMM: An HMM-Based Interactive Map-Matching System[C]// *22nd International Conference on Database Systems for Advanced Applications*. Cham: Springer International Publishing, 2017: 3-18.
- [17] 吴刚,邱煜晶,王国仁.基于隐马尔可夫模型和遗传算法的地图匹配算法[J]. *东北大学学报(自然科学版)*, 2017, 38(4): 472-475.
Wu Gang, Qiu Yujing, Wang Guoren. Map Matching Algorithm Based On Hidden Markov Model and Genetic

- Algorithm[J]. Journal of Northeastern University(Natural Science), 2017, 38(4): 472-475.
- [18] Huang Y, Rao W, Zhang Z, et al. Frequent Pattern-Based Map-Matching on Low Sampling Rate Trajectories[C]// 2018 19th IEEE International Conference on Mobile Data Management. Aalborg, Denmark: IEEE 2018: 266-273.
- [19] Yan Xie, Zhou Kai, Miao Fang, et al. High-Accuracy Off-Line Map-Matching of Trajectory Network Division Based on Weight Adaptation HMM[J]. IEEE Access (S2169-3536), 2020, 8: 7256-7266.
- [20] 吴军. 数学之美[M]. 北京: 人民邮电出版社, 2012: 49-58.
- [21] Antonin Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching[J]. SIGMOD record(S0163-5808), 1984, 14(2): 47-57.