软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(11):2869−2883 [doi: 10.13328/j.cnki.jos.004889]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

大数据环境下移动对象自适应轨迹预测模型*

乔少杰[1], 李天瑞[1], 韩 楠[2], 高云君[3], 元昌安[4], 王晓腾[1], 唐常杰[5]

[1](西南交通大学 信息科学与技术学院,四川 成都 610031)

[2](西南交通大学 生命科学与工程学院,四川 成都 610031)

[3](浙江大学 计算机科学与技术学院,浙江 杭州 310027)

[4](科学计算与智能信息处理广西高校重点实验室(广西师范学院),广西 南宁 530023)

[5](四川大学 计算机学院,四川 成都 610065)

通讯作者: 韩楠, E-mail: hannan@swjtu.edu.cn

摘 要: 已有的轨迹预测算法针对移动对象运动模式,使用数学模型进行交通流模拟,难以对路网中的移动对象进行准确的描述.为了解决这一问题,提出基于隐马尔可夫模型(hidden Markov model,简称 HMM)的自适应轨迹预测模型 SATP(self-adaptive trajectory prediction model based on HMM),对大数据环境下移动对象海量轨迹利用基于密度的聚类方法进行位置密度分区和高效分段处理,减少 HMM 的状态数量.根据输入轨迹自动选取参数组合,避免 HMM 模型中隐状态不连续、状态停留等问题.实验结果表明,SATP 模型在实验中表现出较高的预测准确性,并维持较低的时间开销.针对速度随机改变的移动对象,其平均预测准确率为 84.1%;相同情况下,平均高出朴素预测算法 46.7%.

关键词: 位置大数据;智能交通;轨迹预测;隐马尔可夫模型;自适应

中图法分类号: TP311

中文引用格式: 乔少杰,李天瑞,韩楠,高云君,元昌安,王晓腾,唐常杰.大数据环境下移动对象自适应轨迹预测模型.软件学报, 2015,26(11):2869−2883. http://www.jos.org.cn/1000-9825/4889.htm

英文引用格式: Qiao SJ, Li TR, Han N, Gao YJ, Yuan CA, Wang XT, Tang CJ. Self-Adaptive trajectory prediction model for moving objects in big data environment. Ruan Jian Xue Bao/Journal of Software, 2015,26(11):2869−2883 (in Chinese). http://www.jos.org.cn/1000-9825/4889.htm

Self-Adaptive Trajectory Prediction Model for Moving Objects in Big Data Environment

QIAO Shao-Jie[1], LI Tian-Rui[1], HAN Nan[2], GAO Yun-Jun[3], YUAN Chang-An[4], WANG Xiao-Teng[1], TANG Chang-Jie[5]

[1](School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China)

[2](School of Life Science and Engineering, Southwest Jiaotong University, Chengdu 610031, China)

[3](College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)

[4](Science Computing and Intelligent Information Processing of Guangxi Higher Education Key Laboratory (Guangxi Teachers Education University), Nanning 530023, China)

[5](College of Computer Science, Sichuan University, Chengdu 610065, China)

Abstract: The existing trajectory prediction algorithms focus on the mobility pattern of objects and simulate the traffic flow via mathematical models which are inaccurate at describing network-constraint objects. In order to cope with this problem, a self-adaptive

* 基金项目: 国家自然科学基金(61100045, 61165013); 高等学校博士学科点专项科研基金(20110184120008); 教育部人文社会科学研究规划基金(15YJAZH058); 教育部人文社会科学研究青年基金(14YJCZH046); 中央高校基本科研业务费专项资金(2682013BR023); 四川省教育厅资助科研项目(14ZB0458); 科学计算与智能信息处理广西高校重点实验室开放课题(GXSCIIP201407)

收稿时间: 2015-02-19; 修改时间: 2015-05-11, 2015-07-14; 定稿时间: 2015-08-26

© 中国科学院软件研究所 http://www.jos.org.cn

parameter selection trajectory prediction model based on hidden Markov models (SATP) is proposed. The new model can efficiently cluster and partition location big data, and extract the hidden and observable states by using a density-based clustering approach in order to reduce the number of states in HMM. SATP can automatically select the parameters on the input trajectories and avoid the problems of discontinuous hidden states and state retention. Experimental results demonstrate that the SATP model has high prediction accuracy with less time overhead. The average prediction accuracy of SATP is 84.1% while the moving objects have a random changing speed, which is higher than the Naïve algorithm with an average gap of 46.7%.

Key words: location big data; intelligent transportation; trajectory prediction; hidden Markov model; self-adaptive

随着我国智慧城市建设飞速发展,智能交通系统(intelligent transportation system,简称 ITS)的应用变得日趋普及和大众化.文献[1]指出了大规模数据驱动下 ITS 系统中轨迹的重要性和可能形成的崭新应用.目前,各种交通信息采集技术(如 RFID、无线传感器、视频监控等)已被广泛地运用于高速公路、城市交通的路段和卡口,每天都会收集海量的实时交通数据.同时,问题和困境突显:(1) 智能交通的潜在价值还没有得到有效挖掘;(2) 对交通信息的感知和收集能力有限;(3) 对存在于各个 ITS 中的海量数据无法高效存储及运用、有效分析,缺乏对交通态势的预测和研判能力,对公众的实时交通信息服务很难满足.如何对海量的移动对象轨迹数据进行准确、高效的分析处理及预测,做出即时和正确的交通疏通,进而为有效改善实际交通拥堵状况提供更加智能的基于位置的服务,成为目前亟待解决的关键问题.一个典型应用是:Wang 等人[2]提出了一种基于大规模出租车轨迹数据的交通拥堵可视化分析方法,通过道路速度计算对交通拥堵自动检测.

基于位置的智能服务[3]作为一种新产生的技术,为解决轨迹数据挖掘问题提供了新思路,其目标在于挖掘对象的空间位置所蕴含的价值,分析对象之间的位置关系,从而提供个性化的位置服务.例如,通过记录用户以往的就餐地点,分析其饮食、消费习惯,为用户提供满意的餐厅.为了准确地提供上述基于位置的服务,实时甚至提前获取用户的位置变得尤为重要,因此,移动对象轨迹预测逐渐成为当前研究热点.关于移动对象轨迹预测的研究,为相关科研工作提供了理论基础.例如,对于 GPS 轨迹数据的预处理工作可以辅助提高具体应用的时间效率,提供了更加规范的数据,避免了繁琐的前期工作.此外,轨迹预测工作有利于分析人的行为特征,为社会学家分析个体及群体行为模式提供辅助支持.同时,移动对象轨迹预测具有较高的应用价值.例如,由于出租车数量有限,人们往往难以在高峰期打到车,利用轨迹预测算法可以有效地调度出租车.

位置大数据(location big data)[4]是构成位置社会感知的重要资源,具有相当大的体量,其所使用的数据集在规模和复杂程度上均已达到了“大”数据的层次,代表性实例见表 1.

Table 1 Instances of location big data

表 1 位置大数据实例

实例	描述	来源	采集时间(天)	数据量
手机基站数据	用户通信时刻与位置	美国马萨诸塞	44	100万用户[5]
	用户通信时对应的基站	葡萄牙	365	100万用户[6]
GeoLife数据	基于GPS的轨迹数据	微软亚洲研究院	2007年4月~2012年8月	23 667 828个轨迹点
出租车数据	寻找乘客和空闲出租车	微软亚洲研究院	110	577 000 000个轨迹点
新浪微博数据	用户信息及微博数据	中国新浪门户网站	90	4.9GB

当前,针对大数据环境下的移动对象轨迹预测研究的方法和模型较少,然而,仅一条完整的轨迹往往就包含成百上千的位置点,亟需高效的位置大数据预处理方法.此外,对轨迹进行查询往往需要为位置点构建索引,已有的索引结构,如 DISC-tree[7],均需对路网和轨迹点构建双层树状索引,代价极高.

1 相关工作

国内外研究者针对移动对象频繁模式挖掘及轨迹预测已经展开了相关研究:在使用时空数据挖掘技术挖掘频繁轨迹模式方面,Monreale[8]提出了 WhereNext 方法,从历史轨迹数据中抽取对象的运动模式,借助其表示用户经常频繁访问的地点,同时利用 T-pattern tree 查询发现最佳匹配轨迹;Ying 等人[9]基于地理和轨迹的语义特征预测移动对象下一时刻位置点信息,该方法通过挖掘同类用户的常见行为特性来预测其未来位置.

© 中国科学院软件研究所 http://www.jos.org.cn

马尔可夫链模型在挖掘频繁模式方面应用广泛.Song 等人[10]提出基于状态的移动对象运动模型,运用马尔可夫转移概率解释移动对象在不同状态间的转换.Ishikawa 等人[11]将整个交通路网划分为大小不同的网格并判断移动对象所处位置,以 R-tree 作为辅助索引,使用马尔可夫链描述移动对象在网格间转移的概率,但所提算法主要适用于针对特定区域的分析和精确查询.Asahara[12]借助混合马尔可夫链模型预测行人运动,综合考虑行人的个人特征和历史状态进行预测.实验结果表明,与马尔可夫链和隐马尔可夫模型相比,混合马尔可夫链模型具有较高的预测准确率.Gambs[13]扩展了 Mobility Markov Chain(MMC)模型用于移动对象位置预测,本质是高阶马尔可夫模型,预测精度在 70%~95%之间,但计算开销较大.Qiao 等人[14]将 HMM 应用于移动对象轨迹预测,但未考虑大数据环境下算法的运行时间性能问题.此外,Qiao 等人[15]利用高斯混合模型对移动对象复杂运动模式建模,统计不同运动模式的概率分布,进而实现准确、高效的位置预测.

基于 HMM 的轨迹查询预测算法通常将空间划分为不相交的区域,利用区域代表真实的轨迹点,精简了轨迹数据表示过程,提高了预测计算的速度,适用于位置大数据的分析预测.但是,基于 HMM 的预测模型不可避免地存在答案丢失[16]和精度依赖问题(预测精度依赖于历史轨迹数据).此外,上述算法基于向量空间进行预测,仅适用于不同路口间的路段,当移动对象运动至道路交汇处时,该方法无法预测具体的移动方向.再者,HMM 模型中有很多参数需要人为设置,不同轨迹数据上预测精度差异较大.针对上述方法的不足,本文主要提出了两项创新性工作:(1) 新的位置大数据处理方法,利用基于密度的聚类方法减少 HMM 中状态数量,并对轨迹数据进行分段处理,以提取局部位置数据特征;(2) 新的参数自适应轨迹预测算法,通过计算机轨迹点最小间隔,自适应调整区域划分网格大小和轨迹分段的大小.

2 基本概念及问题描述

针对移动对象位置大数据进行轨迹预测主要步骤包括:(1) 地图预处理,实现网格化分区;(2) 位置轨迹预处理,通过对轨迹数据的简化、抽象和聚集操作,建立高效轨迹预测模型;(3) 局部位置数据特征提取,提取出移动对象的高阶连续运动模式;(4) 位置大数据建模,对轨迹数据的时间和空间维度降维,构建全局模型;(5) 特征关联及预测,运用抽取的轨迹模式借助轨迹预测技术挖掘对象的运动趋势.

本文将轨迹预测问题具体描述为:对于某一特定用户,已知给定轨迹$\{T(t)|t=1,2,3,\ldots,n$,表示不同时间点$\}$和该用户的预测模型参数$\lambda=\{\pi,A,B|\pi$表示初始化概率,A 表示隐状态转移概率矩阵,B 表示隐状态与观察状态间转移概率矩阵$\}$,求解 $n+1$ 时刻移动对象位置 $T(n+1)$.通常,移动对象的轨迹数据由二维坐标点组成,将多个点按时间先后顺序连接起来形成一条完整的动态运动轨迹.轨迹位置大数据除了具有大数据的 4V 特性以外,还具有时序性、不确定性和动态变化性.以 GPS 数据为例,轨迹数据是离散分布在地图上的点,而且体量很大,如果对每条轨迹利用链表进行表达,那么将会占用大量的内存空间,而且不同轨迹之间的相似度将难以描述.因此,为了有效地表示轨迹,本文将平面区域划分成不相交的区域,即,网格.这样就可以使用网格序列代表点序列,完成对轨迹的表示.网格表示法简化了轨迹,优化了内存的使用,同时提高了轨迹相似度计算的效率,但是不可避免地带来了答案丢失问题(具体参见第 4.1 节介绍).

为解决上述问题,本文引入隐马尔可夫模型,首先,通过模型定义完成对轨迹模型的建立;然后,通过解答隐马尔可夫模型的 3 个问题,即,评估问题、解码问题、学习问题,达到轨迹预测的目的.下面给出基于隐马尔可夫模型进行轨迹预测的基本概念,由马尔可夫链引出隐马尔可夫模型.

定义 1(轨迹序列). 给定欧氏空间,轨迹序列 $S=\{s_1,s_2,\ldots,s_n\}$为按时间排序的离散轨迹点,s_i 表示第 i 个轨迹点,$s_i=(x_i,y_i,t_i)$,$1\leqslant i\leqslant n$.

定义 2(预测轨迹序列). 已知轨迹 $S=\{s_1,s_2,\ldots,s_k\}$,利用轨迹预测模型,得到预测轨迹序列 $Tp=\{Tp_1,Tp_2,\ldots,Tp_n\}$,其中,$n>k$,$Tp_i$ 表示第 i 个轨迹点.

定义 3(马尔可夫轨迹链). 已知轨迹随机过程$\Omega=\{X(t),t\in T\}$的状态空间Σ是有限集或可列集,对于 T 内任意 $n+1$ 个轨迹时间序列 $t_1<t_2<t_3<\ldots<t_n<t_{n+1}$ 和Σ内任意 $n+1$ 个状态 $j_1,j_2,j_3,\ldots,j_n,j_{n+1}$,如果条件概率:

$$P\{X(t_{n+1})=j_{n+1}|X(t_1)=j_1,X(t_2)=j_2,X(t_3)=j_3,\ldots,X(t_n)=j_n\}=P\{X(t_{n+1})=j_{n+1}|X(t_n)=j_n\} \tag{1}$$

© 中国科学院软件研究所 http://www.jos.org.cn

恒成立,则称此过程为马尔可夫轨迹链.公式(1)称为马尔可夫性质,或称无后效性.

定义 4(轨迹状态概率). $p_i(t)=P\{X(t)=a_j,t\in T\}$称为马尔可夫轨迹链 $X(t)=a_j$ 的状态概率.

定义 5(轨迹状态转移概率). $p_{ij}(t,t')=P\{X(t')=a_j|X(t)=a_i\}$,称为马尔可夫轨迹链在 $X(t)=a_i$ 的条件下 $X(t')=a_j$ 的状态转移概率;由 p_{ij} 组成的矩阵称为马尔可夫轨迹链的状态转移矩阵,描述由状态 a_i 转移到 a_j 的概率.

定义 6(隐马尔可夫轨迹模型)[14]. 用一个六元组 $H=\{S,H,R,\Pi,A,B\}$来描述,假设 $X(n)$表示轨迹隐状态序列,$O(n)$表示轨迹观测状态序列,则:

- S:表示轨迹序列,$S=\{s_1,s_2,\ldots,s_n\}$为有序的离散轨迹点$\{s_i=(x_i,y_i,t)|1\leqslant i\leqslant n\}$所构成的序列;
- H:轨迹隐状态集合,通过对训练轨迹数据按固定长度分段形成.隐状态由$\{h_i,i=1,2,3,...,N\}$表示,N 表示隐藏状态数量;
- R:轨迹观测状态集合,由空间网格划分单元组成,由$\{o_i,i=1,2,3,...,M\}$表示,M 表示观测状态数量;
- $\Pi=\{\pi_i\}$:轨迹初始状态概率分布,$\pi_i=P(q_1=i)$表示初始化时选择某个状态的概率;
- $A=\{a_{ij}\}$:隐状态转移概率矩阵,其中,$h_{ij}=P\{X(t+1)=\alpha_j|X(t)=a_i\}$,即,下层 Markov 轨迹链状态转移矩阵;
- $B=\{b_{ik}\}$:轨迹观测状态与隐状态转移概率矩阵,也称为混淆矩阵,其中,$b_{ik}=P\{O(t)=o_k|X(t)=h_i\}$,描述某个时刻由隐藏状态 a_i 得到观测状态 o_k 的概率.

隐状态转移矩阵和混淆矩阵与时间无关,即,当系统演化时,A 和 B 并不随时间改变.当 N 和 M 固定时,本文使用$\lambda=\{\pi,A,B\}$表示模型的参数.

图 1 是隐马尔可夫轨迹模型的一个实例,椭圆代表训练轨迹通过划分后形成的隐状态,二维移动空间平面网格划分单元作为观测状态.其中,轨迹隐状态集合 $H=\{s_1,s_2,s_3,s_4,s_5\}$,状态数 $N=5$,轨迹观测状态集合 $R=\{g_1,g_2,g_3,g_4,g_5,g_6,g_7,g_8,g_9\}$,$M=9$.隐状态转移概率矩阵和混淆矩阵如下:

$$A=\{h_{ij}\}=\begin{Bmatrix}0 & 0.3 & 0.7 & 0 & 0\\ 0 & 0 & 0.4 & 0.2 & 0.4\\ 0 & 0 & 0 & 0.45 & 0.55\\ 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0\end{Bmatrix},B=\{b_{ik}\}=\begin{Bmatrix}0.4 & 0 & 0 & 0.6 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0\\ 0 & 0 & 0 & 0 & 0.35 & 0.6 & 0 & 0.05 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0.7\end{Bmatrix}.$$

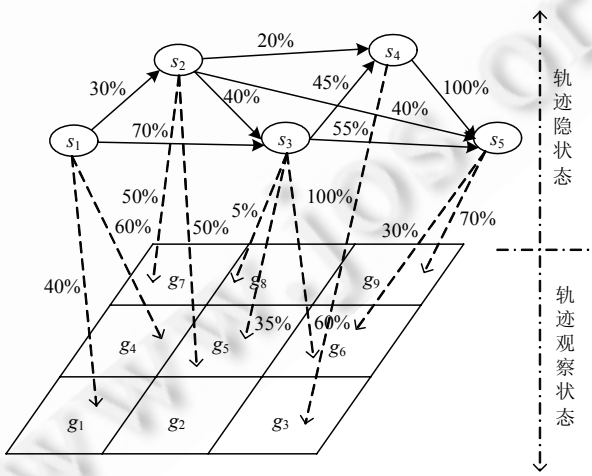


Fig.1 Instance of trajectory hidden Markov model

图 1 隐马尔可夫轨迹模型示例

为了计算矩阵 A,需要将一条完整轨迹按固定长度分段,检查一段轨迹上的所有点 p_k 是否都落在 S 中:如果“是”,则记下该隐状态的标识符;否则,记为“空”(用*表示).因此,子轨迹能够表示为一个状态序列如 $s_1s_3{**}s_5$.然后,根据该序列计算轨迹状态转移概率.矩阵 B 的计算通过检查 p_k 是否属于 g_i 和 s_i 或只属于 g_i.在使用隐马尔可

© 中国科学院软件研究所 http://www.jos.org.cn

夫模型对位置大数据进行建模前,需要对轨迹数据进行预处理工作,包括去噪、分段等处理,使轨迹数据得以化简,进而更加符合大数据环境下隐马尔可夫模型建模要求,下一节将详细阐述.

3 位置大数据预处理

面对体量和规模巨大、时空关系复杂的位置大数据,常常需要对轨迹数据进行预处理,本文提出新型大数据环境下位置密度分区和局部位置数据特征提取方法.位置密度分区通过抽取对象的特征,将特征相似的多个对象组成类,达到对数据分类与筛选的目的,同一个类中的成员对象具有相似的属性.DBSCAN(density-based spatial clustering of applications with noise)算法[17]是基于密度对数据进行处理的经典方法.轨迹数据在时空上的分布具有不确定性,某些区域轨迹点较密集,而一些区域较分散,因此特别适合于利用基于密度的分区方法对轨迹数据进行预处理.位置大数据分析通常需要基于路网或地图数据展开,因此需要将连续的空间地图进行离散化处理,划分为多个区域,依位置密度分布进行分区是一种较好的策略.

本文对 DBSCAN 聚类算法进行改进,用于对位置大数据进行分区划分.此外,位置大数据包含了大量信息,用途广泛,在使用这些数据之前,有诸多问题需要解决:(1) 连续的轨迹数据由定位终端按一定时间间隔采样而成,由于移动对象速度变化、定位设备精确度不高等因素的影响,生成的轨迹数据不完全符合真实情况;(2) 由于数据采集设备自身的不稳定性,采集到的数据集中往往会包含一些噪声点.

3.1 位置密度分区

本节使用基于密度的方法对位置大数据分区,将轨迹点与由信号不稳定等因素产生的噪声点区分,其本质是轨迹聚类,输入参数包括邻域半径ε和最小轨迹点数θ.下面给出位置密度分区算法中的主要概念.

定义 7(*ε*邻域). 给定轨迹点 p,将半径为ε内的区域称为该轨迹点的ε邻域.

定义 8(核心对象). 给定轨迹点 p 的ε邻域内轨迹点数量大于θ,则对象 p 被称为核心对象.

位置密度分区法的基本思想是:通过遍历位置大数据中每个轨迹点的邻域来生成簇.假设一个轨迹点 p 的邻域内包含多于θ个轨迹点,则创建一个新簇,将 p 作为该簇的核心对象.然后,递归遍历核心对象直接密度可达的对象,过程中包含簇合并操作.当没有新的点可以添加到任何簇时,过程结束.算法如下所示[14]:

算法 1. 基于密度的轨迹分区算法——TraCluster.

输入:位置大数据集合 S,聚簇半径ε,最少轨迹点数目θ.

输出:达到聚类密度要求的簇集合 $C=\{C_1,C_2,\ldots,C_n\}$.

```
1.  n=0;                          //初始化簇的个数为 0
2.  for each unvisited point p in S
3.     report p as visited;       //将 p 标记为已访问
4.     R=Neighbours(p,ε);
5.     if size(R)<θ then
6.        report p as noise;      //如果满足 size(R)<θ,则将 p 标记为噪声
7.     else
8.        create a new cluster C_k;   //建立新簇 C_k
9.        ExpandCluster(p,N,C_k,ε,θ);
10.    end if
11. end for
```

在位置密度分区过程中,遍历 S 中所有轨迹点.如算法 1 所示.

- 第 1 行初始化簇个数;
- 第 2 行开始遍历轨迹点;
- 第 3 行将轨迹点 p 标记为已访问;
- 第 4 行~第 11 行计算轨迹点 p 与其他所有轨迹点的距离,将距离小于ε的轨迹点存入集合 R 中,如果 R

© 中国科学院软件研究所 http://www.jos.org.cn

小于θ,则标记 p 为噪声;否则,以 p 为核心建立新簇,并调用 ExpandCluster 算法递归访问 R 中轨迹点. ExpandCluster 算法如下所示:

算法 2. *ExpandCluster*($p,R,C_k,\varepsilon,\theta$).

输入:p 的邻域 R,簇 C_k,半径ε,最少轨迹点数目θ.

输出:所有达到密度要求的簇.

```
1. add p to cluster C_k;                    //首先将核心点加入簇 C_k
2. for each point p' in R
3.     report p' as visited;
4.     R'=Neighbours(p',ε);                 //对 R 邻域内的所有点在进行半径检查
5.     if size(R')≥θ then
6.         add p' to cluster C_k;           //将 p'加入簇 C_k
7.         ExpandCluster(p',R',C_k,ε,θ);
```

在 ExpandCluster 算法中,深度优先访问 R 中轨迹点:

- 首先,将点 p 加入簇 C_k 中(第 1 行);
- 将 p 标记为已访问(第 2 行、第 3 行);
- 判断 R 中点 p'是否为核心对象,如果是,则将 p'的加入到 C_k 中(第 4 行~第 6 行);执行递归运算深度优先判断 p'的ε邻域(第 7 行).

借助空间索引技术 DISC-tree[7],本文所提位置密度分区算法时间复杂度为 $O(n\log n)$,n 表示数据集中对象的数目.

3.2 局部位置数据特征提取

为了提高模型对位置大数据预处理的效率,需要对轨迹数据进行分段处理,轨迹分片称为“段”.本文提出轨迹分段算法 TraSegment[14],用于提取局部位置数据特征.轨迹分段算法仍然使用ε邻域中半径ε作为参数.如算法 3 所示,轨迹分段的基本思想是:

(1) 进行初始化工作(第 1 行).

(2) 遍历 *TraList* 中的轨迹点(第 2 行),判断其是否已经访问过:如果访问过,则跳过;否则,设置其段 *id* 为 c,并标记为已访问(第 3 行~第 8 行).从 $j=i+1$ 点开始遍历,如果点 i 与点 j 的距离小于ε,则标记点 j 的段 *id* 为 c,并标记为已访问(第 9 行~第 17 行).

算法 3. 局部位置数据特征提取算法——TraSegment.

输入:包含 n 个位置点的轨迹链表 *TraList*,分段半径ε.

输出:所有达到分割要求的簇.

```
1.  c=1;                                    //设置段 id 从 1 开始
2.  for i=0 to TraList.length               //遍历轨迹点
3.      if TraList[i].visit==true
4.          continue;
5.      else
6.          TraList[i].id=c;
7.          TraList[i].visit=true;          //将 p 标记为已访问
8.      end if
9.      for j=i+1 to TraList.length
10.         if TraList[i].visit==true
11.             continue;
12.         end if
```

© 中国科学院软件研究所 http://www.jos.org.cn

```
        if Distance(TraList[i],TraList[j])<ε
            TraList[j].id=c;
            TraList[j].visit=true;              //将 p 标记为已访问
        end if
    end for
    c++;
end for
```

轨迹分段的效果如图 2 所示,可以看到:通过应用分段算法,连续的轨迹被划分成为不同的片段.轨迹分段主要作用是:(1) 进一步进行位置大数据预处理操作,提高预测算法运行效率;(2) 位置大数据中局部位置数据的特征提取,提取轨迹预测模型中包含的隐状态.

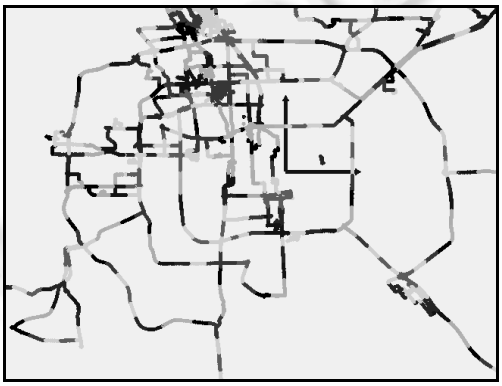


Fig.2 Results of trajectory segmentation

图 2 轨迹分段结果

4 基于 HMM 的参数自适应轨迹预测算法

4.1 工作原理

本文提出的基于 HMM 的参数自适应轨迹预测算法以隐马尔可夫模型为基础,根据轨迹数据特点,抽象出与 HMM 模型相对应的轨迹隐状态和观测状态,通过解决 HMM 模型的 3 个基本问题完成轨迹预测.

轨迹模型建立过程中,首先使用轨迹分段算法对位置大数据进行分段,将用于训练的轨迹划分成不同的段,用$\{C_i,i=1,2,3,...,M\}$表示.同时,为简化轨迹,仍采用平面区域划分方法将轨迹表示成为网格序列,用$\{g_i,i=1,2,3,...,N\}$表示.如图 3 所示,轨迹 T_1 被表示为$\{g_1,g_2,g_6,g_{10}\}$和$\{C_1,C_2,C_3\}$,前者为网格序列表示,后者为段序列表示,分别对应 HMM 中的观测状态和隐状态;M 和 N 分别为隐状态和观测状态数量.

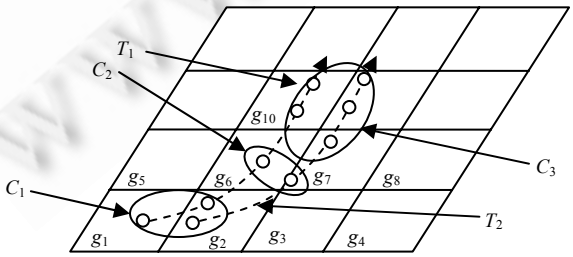


Fig.3 Answer-Loss problem

图 3 答案丢失问题

© 中国科学院软件研究所 http://www.jos.org.cn

由图 3 可知,将移动空间按照网格划分后,原本相近的两条轨迹被划分到了不同的网格单元中.在完整的空间区域中,可以通过计算两条轨迹中轨迹点的欧氏距离来判断轨迹相似度.但是在经网格划分后的空间中,欧式距离这一参数将不再具有意义,取而代之的是网格的邻近程度.轨迹 T_1 和轨迹 T_2 在网格划分下相似度较低,因为两条轨迹分属于不同的网格,邻近程度较低,使得原本相似度较高的两条轨迹在网格划分下表现出较低相似度,这就是答案丢失问题.为了解决这一问题,引入 HMM 模型,轨迹 T_1 和 T_2 同属于$\{C_1,C_2,C_3\}$这一段序列.在 HMM 模型中,使用段序列表示轨迹,通过判断不同轨迹是否属于相同的段来计算轨迹相似度,这样就避免了直接使用网格序列表示轨迹带来的答案丢失问题.

应用所提出的轨迹模型,当输入轨迹由网格序列表示时,可以根据 HMM 的解码问题求解出对应的隐状态序列,即得到由段序列表示的轨迹.解码问题是 HMM 中的一个重要问题.在很多情况下,人们对隐状态更感兴趣,因为其包含了一些不能被直接观察到的有价值的信息.对于轨迹预测模型而言,当已知一个网格表示的轨迹序列 $R_m(t)=\{g_i|i=1,2,3,...,N\}$时,如何根据已知的$\lambda=\{\pi,A,B\}$求出对应最可能的段序列 $C_n(t)=\{C_i|i=1,2,3,...,M\}$,这一问题就可以转化为 HMM 中的解码问题,也是本文解决的一个关键问题.已知:

$$P(R\mid\lambda)=\underset{C_1,C_2,...,C_M}{\mathrm{Max}}\ P(R\mid C_i,\lambda)P(C_i\mid\lambda) \tag{2}$$

其中,R 表示输入轨迹序列,C_i 表示 R 可能对应的某一段序列.为了求使得 $P(R|\lambda)$取最大值时 C_i 的值,需要通过枚举所有可能的段序列,带入模型计算其在给定网格序列情况下出现的概率.这时候,使用维特比算法 Viterbi[18],根据轨迹观测序列得到概率最大的隐状态序列.

在进行轨迹预测时,首先将输入的轨迹投影到网格划分的区域中.如图 4 所示,将轨迹 l_1 转换为网格序列$\{g_1,g_2,g_6,g_{10}\}$表示,然后使用训练轨迹数据建立预测模型,应用评估问题,针对不同轨迹进行预测,即:将轨迹带入模型计算每条轨迹出现的概率,取概率最大轨迹作为预测轨迹 $T(t)$,$t=n+1$ 时刻对应的隐状态,并用轨迹分段的中心点代表预测点.

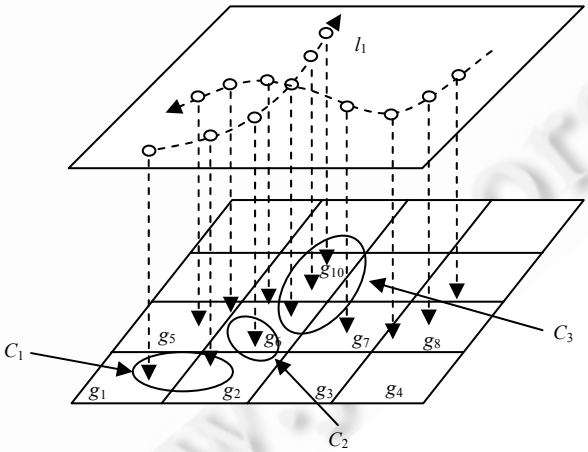


Fig.4 Example of trajectory projection

图 4 轨迹投影举例

针对评估问题,其具体描述为:已知模型参数$\lambda=\{\pi,A,B\}$,如何计算某一给定网格表示的轨迹序列 $R_m(t)=\{g_i|i=1,2,3,...,N\}$在模型中出现的概率,即 $P(O|\lambda)$.从 HMM 模型的定义中可以知,对于某一给定的观测序列,其对应的隐状态序列存在 N^L 个,其中,L 为给定观察序列的长度.假设网格序列 R 对应的段序列集为 $C_i=(C_1,C_2,\ldots,C_T)$,$T=N^L$,则:

$$P(R\mid\lambda)=\sum_{C_1,C_2,...,C_T}P(R\mid C,\lambda)P(C\mid\lambda) \tag{3}$$

上式的计算时间复杂度为 $O(N^L)$.为了高效地解决轨迹预测的评估问题,本文使用前向算法 Forward[18].

© 中国科学院软件研究所 http://www.jos.org.cn

4.2 自适应参数选择算法

基于HMM的轨迹预测模型的建立,与轨迹所在区域的大小密切相关.轨迹预测模型的精确度受3个主要因素的影响,分别是 HMM 模型参数$\lambda=\{\pi,A,B\}$、区域划分网格大小 *gridSize*、轨迹分段的大小 *segSize*.在实际应用中,用于预测的已知轨迹可能并不符合预测模型的理想输入.由于移动对象的运动速度可以是随机变化的,所以任意两个轨迹点的间隔大小并不一致,这就导致了由输入产生的隐状态链不连续的问题.为了解决这一问题,本文首先计算得到轨迹点最小间隔 min*Dis*,然后对包含轨迹的区域进行整体缩放,使得 min*Dis* 小于模型的聚类最小间隔.这样得到的轨迹最小间隔满足了模型要求,但是缩放使其他间隔加大,所以要通过添加轨迹中间点填补空缺轨迹.上述过程被称为参数自适应选择,如算法 4 所示,基本思想为:第 1 行根据输入位置点计算轨迹点最小间隔;第 2 行、第 3 行将参数 *segSize*,*gridSize* 设置为最小间隔;第 4 行~第 10 行判断轨迹间隔是否过大,如超过要求,则应用线性插值法插入轨迹点,进行轨迹补全;第 11 行、第 12 行将补全后的位置点投影到网格上表示;第 13 行返回结果.将算法 4 得到的网格序列作为输入数据带入第 4.4 节介绍的轨迹预测算法,便可以获得一条连续的预测轨迹序列.自适应参数选择算法主要解决真实应用中用于预测的轨迹间隔随机变化导致预测准确率下降的问题,根据输入轨迹自动选取参数组合,避免了隐状态不连续、状态停留问题.

算法 4. 自适应参数选择算法——*ParaSelection*(*S*).

输入:任意一条轨迹序列 $S=\{s_1,s_2,\ldots,s_n\}$.

输出:轨迹在网格上的投影点集.

```
1.  min=MinDistance(S);                    //取输入轨迹点最小间隔
2.  segSize=min;
3.  gridSize=min;
4.  distance=0;
5.  for (i=0; i<S.length−1; i++)
6.      distance=Distance(S[i],S[i+1]);
7.      if (distance>segSize×2)
8.          Insert();
9.      end if
10. end for
11. for each p in S
        grid.add(TransfromIntoGrid(p));    //格式化网格序列
12. end for
13. return grid;
```

4.3 模型存在问题解决[14]

(1) 隐状态链不连续问题解决策略

HMTP 模型中包含一条马尔可夫链,用来描述段的状态转移.如图 5 所示,圆点表示训练数据,三角形表示预测时的前 n 个轨迹点.段簇的转移代表着移动对象的轨迹序列,当网格大小不大于段的大小时,预测所需的轨迹点所处的网格对应轨迹形成的段,因此,从图 5 中可以发现,轨迹对应的隐状态是连续的.

然而当网格过大时,模型对于轨迹的描述就变得不尽如人意了.如图 6 所示,网格尺寸相对于段过大,当判别轨迹点 p 对应的隐状态时出现偏差.因为对于 p 所处的网格 R 来说,其对应的概率最大段为 C,因此,落入 R 中的轨迹点都将被视为属于 C,而忽略 p 本应属于的段,导致隐状态链不连续.

由于预测算法使用前向算法进行计算,在状态转移矩阵时,不连续的隐状态之间的转移概率为 0,因此前向算法的输出结果必为 0,导致预测失败.解决这一问题的一种方法是:在 HMM 建立过程中分梯度多次使用历史数据,即,以周期$\{T|T>gridSize\}$为间隔提取轨迹点,与连续轨迹点一同进行状态转移矩阵生成计算,得到的矩阵

© 中国科学院软件研究所 http://www.jos.org.cn

中不连续的两个状态的转移概率.这样,前向算法的计算得以进行.

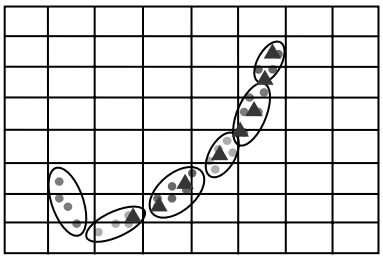


Fig.5 Trajectory state transfer

图 5 轨迹状态转移

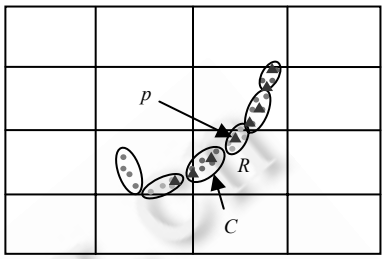


Fig.6 Prediction bias caused by zooming in the grid

图 6 放大网格导致的预测偏差

(2) 状态停留问题解决策略

状态停留问题指的是前 n 个轨迹点中,存在多个连续点属于同一段的情况.由状态转移矩阵的定义可知,$p_{ii}=0,1\leqslant i\leqslant N$,导致预测失败.因此,可以通过实验获得 p_{ii} 的经验值,使得 $0<p_{ii}<\max\{p_{ij}\},1\leqslant i\leqslant N,1\leqslant j\leqslant N$.

4.4 基于HMM的自适应轨迹预测算法

计算出 HMM 模型的具体参数$\lambda=\{\pi,A,B\}$以后,下一步进行轨迹预测.

预测问题具体描述为:已知给定轨迹$\{T(t),t=1,2,3,\ldots,n\}$和$\lambda$,求解 $T(n+1)$.对此,可以通过枚举下一轨迹点可能属于的网格,组成轨迹$\{T(t),t=1,2,3,\ldots,n+1\}$,带入评估问题,求解出概率最大的预测轨迹,其 $T(n+1)$即为预测点.但是这种方法求得的预测点为网格,预测精度较低.因此,改进评估问题中的前向算法,使得输出为概率最大的段中心点,这样得到的预测点精度相比网格有很大提高,轨迹预测算法如下所示.

算法 5. 基于 HMM 的自适应轨迹预测算法.

输入:任意一条轨迹序列 $S=\{s_1,s_2,\ldots,s_n\}$.

输出:预测点坐标$(x_{n+1}, y_{n+1}, t_{n+1})$.

```
1.  TraCluster(S);                    //基于密度的轨迹分区预处理
2.  TraSegment(S);                    //轨迹分段
3.  ParaSelection(S);                 //初始化操作
4.  Transform(S);                     //将位置大数据转化为网格链表示
5.  if (S.length≤0)
6.     return false;
7.  Forward(S,tail);                  //使用前向算法计算
8.  Viterbi(S,bestSeq);               //使用维特比算法计算输入轨迹对应的隐状态
9.  for i=0 to m                      //遍历预测时刻隐状态 i,m 表示隐状态转移概率矩阵维度
10.    if (i in bestSeq)              //防止位置点倒退,去除已遍历过的段
11.       continue;
12.    for j=0 to m                   //遍历 tail 中隐状态 j
13.       next[i]+=tail[j]×M[j,i];    //计算由状态 j 转换为 i 的概率,M 表示隐状态转移概率矩阵
14.       max=i while next is Maximum; //i 值即为预测的隐状态的段号
15. return (x_{n+1},y_{n+1},t_{n+1});  //第 i 个段中心点的坐标
```

轨迹预测算法基本思想是:

- 针对每条轨迹序列,第 1 行进行位置密度分区操作,第 2 行进行局部位置数据特征提取,第 3 行调用自适应参数选择算法;

© 中国科学院软件研究所 http://www.jos.org.cn

- 第 4 行进行轨迹投影将输入原始轨迹转化为由网格表示的网格链;
- 第 5 行、第 6 行判断输入轨迹长度是否符合要求;
- 第 7 行、第 8 行分别使用前向算法和维特比算法得到前向变量的最后一列 *tail*,用以保存当前状态到各个状态的状态转移概率和最佳隐状态序列 *bestSeq*;
- 第 10 行、第 11 行判断隐状态是否已经在 *bestSeq* 中出现,如果出现,则舍弃该隐状态;
- 第 12 行~第 14 行计算从 *S* 序列对应的最后一个隐状态转移到所有遍历的隐状态的概率;
- 当概率取最大时,第 15 行返回对应的隐状态的中心点,作为预测的目标点.

5 实验与性能分析

5.1 实验环境及数据集描述

本实验所使用的位置大数据集来自微软亚洲研究院的 GeoLife 项目[19],由 182 个用户 5 年间 11 129 天的 GPS 轨迹数据组成.数据集包含 17 621 条轨迹数据,包含 23 667 828 个轨迹点,总长度达 1 292 951 公里.数据分布在北京市主城区的各条街道上,很好地反映了移动对象的真实运动行为.这些 GPS 数据由不同的定位装置收集,包括 GPS 接收机、智能手机等.文中算法利用 C#程序设计语言实现,使用 Microsoft Visual Studio 2008 开发环境,数据库为 SQL Server 2008.实验硬件平台为:Intel(R) Core(TM) 2 Duo P8700 2.53GHz CPU,2GB 内存,操作系统平台为 Windows 7.实验中实现了基于 HMM 的自适应轨迹预测算法 SATP、朴素的基于 HMM 未考虑参数自适应选择的预测算法 Naïve、基于时间连续贝叶斯网络的轨迹预测算法 PutMode[20]、基于高斯混合模型的轨迹预测算法 GMTP[15].为方便比较不同算法的优劣,性能评价指标定义如下:

定义 9(预测命中率). 已知轨迹序列 $T=\{T_1,T_2,\ldots,T_k\}$,预测轨迹序列 $Tp=\{Tp_1,Tp_2,\ldots,Tp_n\}$,$k<n$,$dist(p,q)$表示空间中 p 和 q 两点间的欧氏距离,η表示距离阈值,则 $dist(T_i,Tp_i)<\eta$时表示预测命中,定义如下:

$$H(T_i,Tp_i)=\begin{cases}1, & (dist(T_i,TP_i)<\eta)\\ 0, & (dist(T_i,TP_i)>\eta)\end{cases} \tag{4}$$

定义 10(预测准确率). 已知轨迹序列 T,预测轨迹序列 Tp,则预测准确率定义为

$$Accuracy=\frac{\sum_{i=1}^{n}H(T_i,Tp_i)}{|Tp|} \tag{5}$$

其中,$|Tp|$代表预测轨迹序列的长度.

定义 11(预测偏离度). 已知轨迹序列 T,预测轨迹序列 Tp,则预测偏离度定义为

$$Deviation=\frac{\sum_{i=1}^{n}dist(T_i,Tp_i)}{|Tp|} \tag{6}$$

5.2 速度随机变化算法性能分析

由于实际应用中移动对象的移动速度随机变化,不是恒定不变的,导致轨迹间隔大小不一,本文提出了基于 HMM 的自适应轨迹预测算法 SATP.下面对比在移动对象速度随机变化,即,轨迹间隔大小不一的情况下,不同算法的预测效果.实验采用 8 组不同轨迹数据集,每组数据包含 2.5×10^6 个轨迹点作为训练数据.

通过对比各种算法在不同位置大数据集上的预测精度可以发现(如图 7 所示):

(1) 与 Naïve 和 PutMode 算法相比,SATP 模型在不同数据集上进行轨迹预测具有较高的预测准确率,平均高出 Naïve 算法 46.7%,高出 PutMode 算法 27.2%,高出 GMTP 算法 10.3%.因为 SATP 模型总结了基于 HMM 的轨迹预测模型中隐状态的特点,针对朴素算法参数恒定、无法灵活应对速度随机变化的移动对象这一不足提出改进,对输入轨迹动态判断,分析移动对象的速度特征,在每次预测前调整网格大小和分段大小这两个参数,使得模型更加适合对移动对象当前和未来位置的预测.

© 中国科学院软件研究所 http://www.jos.org.cn

(2) 当面对不同轨迹数据时,Naïve,PutMode 和 GMTP 算法的预测准确率不稳定,时高时低,而 SATP 模型相对稳定,准确率维持在较高水平.主要是因为其可以针对不同数据自适应选择参数,更好地适应不同轨迹数据.

图 8 给出了不同预测算法在 8 组实验数据上的预测偏离度:当预测轨迹由速度随机变化的移动对象产生时, SATP 模型的参数也相应改变,其中分段大小这一参数也随之改变.根据预测偏离度的定义可知,分段大小会对偏离度产生一定影响,因此,SATP 模型的预测偏离度低于 Naïve 和 PutMode 算法.此外可以发现,GMTP 算法的预测偏离度略高于 SATP.其原因在于:GMTP 算法利用高斯过程回归预测移动对象最可能的运动轨迹,可以较为准确地预测对象未来位置点.此外可以发现,PutMode 算法的预测偏差最高.原因在于其采用不确定性轨迹圆柱的圆心代表预测点,位置预测的精度较低,因此会产生较大的预测偏差.

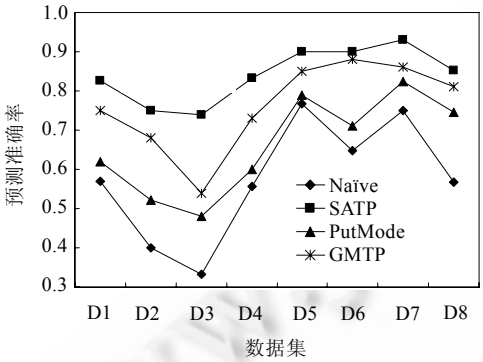


Fig.7 Accuracy comparison in randomly changing speed

图 7 变速情况下预测准确率比较

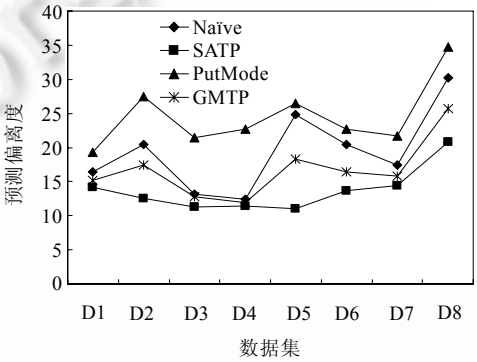


Fig.8 Deviation comparison in randomly changing speed

图 8 变速情况下预测偏离度比较

5.3 速度小范围变化算法性能分析

对于速度变化较低的移动对象,通过实验对比 4 种算法的预测效果.实验结果如图 9 和图 10 所示.通过观察可以发现:在同样的数据集上,尽管 SATP 和 Naïve 算法的预测准确率接近,但是 SATP 模型的预测偏离度低于 Naïve 算法,轨迹预测的精度明显好于 PutMode 和 GMTP 算法.实验结果表明:对速度相对恒定的移动对象进行轨迹预测时,SATP 和 Naïve 算法的准确率相近.因为匀速运动的移动对象的轨迹间隔变化较小,不需要更改分段大小或网格大小就可以获得较高的预测准确率.由于 SATP 模型动态改变分段大小,使得在距离上的划分粒度更为精细,因此预测点与实际点的距离较小,预测偏离误差低于恒参的 Naïve 算法.

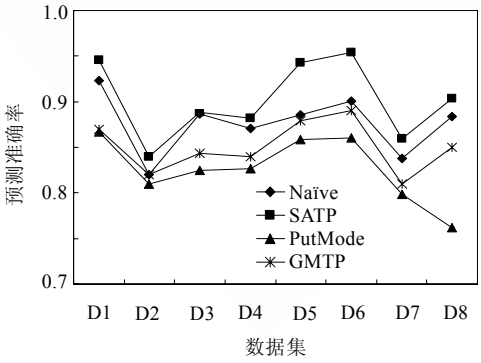


Fig.9 Accuracy comparison in constant speed

图 9 恒定速度情况下预测准确率比较

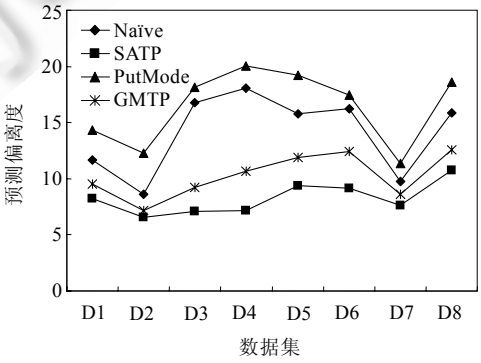


Fig.10 Deviation comparison in constant speed

图 10 恒定速度情况下预测偏离度比较

© 中国科学院软件研究所 http://www.jos.org.cn

综上所述,SATP 模型综合考虑移动对象真实状态特征,根据输入数据自主调整模型参数,改善预测效果.与未考虑参数自适应调节的 Naïve,PutMode 和 GMTP 算法相比,该模型在预测准确率和偏差上具有明显的优势.

5.4 模型建立和预测时间效率分析

在实时预测系统中,轨迹预测模型的建立与预测效率十分重要,为了验证轨迹预测模型建立和预测的时间效率,选取 10 000 条训练轨迹进行实验,其中每条轨迹包含约 10 万个轨迹点.本节比较 4 种算法的运行时间性能,结果如图 11 所示.

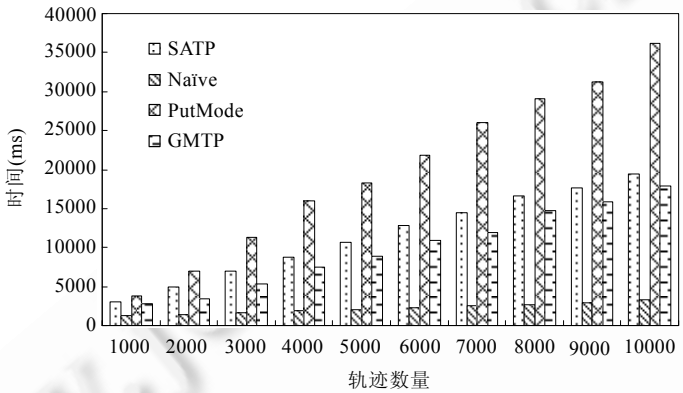


Fig.11 Execution time of creating TP model and prediction

图 11 轨迹预测模型建立及预测时间

实验结果表明:

(1) GMTP 算法略优于 SATP 算法.与 SATP 和 PutMode 算法相比,Naïve 算法模型建立及预测时间较低,随训练轨迹增多而增大,保持线性关系.当训练轨迹数量达到 10 000 条(大约 10^9 个轨迹点)时,模型建立及预测时间为 3s,实时性较好.

(2) SATP 模型的建立及预测时间与 Naïve 算法相比有所增加,但总体消耗保持在可以接受的范围内,而且与训练轨迹数量成线性关系.SATP 模型的建立及预测时间稍高于 Naïve 算法的原因在于:轨迹预处理阶段,由于模型参数动态变化,需要自适应选择参数,导致需要进行区域划分和轨迹分段,耗费了时间;反观 Naïve 算法,由于参数恒定,所以模型建立一次性完成,只进行一次轨迹分段与区域划分,因此时间开销较小.实验中为了展示算法的特点,选取极端情况下的数据进行测试,而真实情况下移动对象的速度变化相对缓慢.也就是说,SATP 模型的参数变化频率较低,因此时间开销相对较小.与 PutMode 算法相比,其运行时间平均降低 38.9%.其原因在于:PutMode 算法进行预测时需要花费大量时间构建时间连续贝叶斯网络;此外,其轨迹聚类操作的时间开销较大,不适合位置大数据的预测.

(3) 在算法的真实应用中,用于训练的轨迹数据远小于实验数据,在位置大数据环境下(1 万条轨迹,大约 100 万个轨迹点),SATP 模型建立和预测时间均保持在 20s 以下,具有良好的实时性.

6 结论与展望

本文旨在研究位置大数据环境下的新型高效、准确的轨迹预测方法,为了对位置大数据进行高效预处理,提出了基于密度的聚类方法用于位置密度分区和轨迹分段局部位置数据特征提取方法.为了适应移动对象速度动态变化的真实运动场景,提出了一种自适应参数选择算法,基于该算法设计实现了新型轨迹预测模型.位置大数据集上的实验,验证了本文所提自适应轨迹预测方法的时间性能优势及预测的精准性.最后,在基于 HMM 的轨迹预测模型基础上,设计实现了一个移动对象轨迹预测系统,综合运用前文所述理论依据,提供了对预测结果直观的展示,系统详细描述请参见 http://userweb.swjtu.edu.cn/Userweb/qiaoshaojie/demo2.htm.

© 中国科学院软件研究所 http://www.jos.org.cn

未来的研究工作包括:(1) 充分考虑客观因素对移动对象位置预测的影响,如,红绿灯、天气等因素,提高预测算法对环境因素的自适应性;(2) 为了进一步提高算法运行效率,将本文提出的轨迹预测算法并行化,将算法移植到 Hadoop 平台,为公安部门提供实时交通流监控和预测,辅助智能交通控制.

References:

[1] Zhang JP, Wang FY, Wang KF, Lin WH, Xu X, Chen C. Data-Driven intelligent transportation systems: A survey. IEEE Trans. on Intelligent Transportation Systems, 2011,12(4):1624−1639. [doi: 10.1109/TITS.2011.2158001]

[2] Wang ZC, Lu M, Yuan XR, Zhang JP, Wetering H. Visual traffic jam analysis based on trajectory data. IEEE Trans. on Visualization and Computer Graphics, 2013,19(12):2159−2168. [doi: 10.1109/TVCG.2013.228]

[3] Meng XF, Ding ZM. Mobile Data Management: Concepts and Techniques. Beijing: Tsinghua University Press, 2009. 185−200 (in Chinese).

[4] Guo C, Liu JN, Fang Y, Luo M, Cui JS. Value extraction and collaborative mining methods for location big data. Ruan Jian Xue Bao/Journal of Software, 2014,25(4):713−730 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4570.htm [doi: 10.13328/j.cnki.jos.004570]

[5] Calabrese F, Pereira FC, Francisco C, Di Lorenzo G, Liu L, Ratti C. The geography of taste: Analyzing cell-phone mobility and social events. In: Proc. of the 8th Int'l Conf. on Pervasive Computing. Springer-Verlag, 2010. 22−37. [doi: 10.1007/978-3-642-12654-3_2]

[6] Calabrese F, Smoreda Z, Blondel VD, Ratti C. Interplay between telecommunications and face-to-face interactions: A study using mobile phone data. PLoS ONE, 2011,6(7):e20814. [doi: 10.1371/journal.pone.0020814]

[7] Qiao SJ, Han N, Wang C, Zhu F, Tang CJ. A two-tiered dynamic index structure of moving objects based on constrained networks. Chinese Journal of Computers, 2014,37(9):1947−1958 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2014.01947]

[8] Monreale A, Pinelli F, Trasarti R, Giannotti F. WhereNext: A location predictor on trajectory pattern mining. In: Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2009. 637−646. [doi: 10.1145/1557019.1557091]

[9] Ying JJ, Lee W, Weng T, Tseng VS. Semantic trajectory mining for location prediction. In: Proc. of the 19th ACM SIGSPATIAL Int'l Conf. on Advances in Geographic Information Systems. New York: ACM Press, 2011. 34−43. [doi: 10.1145/2093973.2093980]

[10] Song MB, Ryu JH, Lee SK, Hwang CS. Considering mobility patterns in moving objects database. In: Proc. of the 2003 Int'l Conf. on Parallel Processing. Washington: IEEE, 2003. 597−604. [doi: 10.1109/ICPP.2003.1240628]

[11] Ishikawa Y, Tsukamoto Y, Kitagawa H. Extracting mobility statistics from indexed spatio-temporal datasets. In: Proc. of the 2nd Int'l Workshop on Spatio-Temporal Database Management. 2004. 9−16.

[12] Asahara A, Maruyama K, Sato A, Seto K. Pedestrian-Movement prediction based on mixed Markov-chain model. In: Proc. of the 19th ACM SIGSPATIAL Int'l Conf. on Advances in Geographic Information Systems. New York: ACM Press, 2011. 25−33. [doi: 10.1145/2093973.2093979]

[13] Gambs S, Killijian M, Cortez DP, Miguel N. Next place prediction using mobility Markov chains. In: Proc. of the 1st Workshop on Measurement, Privacy, and Mobility. New York: ACM Press, 2012. 1−6. [doi: 10.1145/2181196.2181199]

[14] Qiao SJ, Shen DY, Wang XT, Han N, Zhu W. A self-adaptive parameter selection trajectory prediction approach via hidden Markov models. IEEE Trans. on Intelligent Transportation Systems, 2015,16(1):284−296. [doi: 10.1109/TITS.2014.2331758]

[15] Qiao SJ, Jin K, Han N, Tang CJ, Gesangduoji, Gutierrez LA. Trajectory prediction algorithm based on Gaussian mixture model. Ruan Jian Xue Bao/Journal of Software, 2015,26(5):1048−1063 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4796.htm [doi: 10.13328/j.cnki.jos.004796]

[16] Jensen CS, Lin D, Ooi BC, Zhang R. Effective density queries on continuously moving objects. In: Proc. of the 22nd Int'l Conf. on Data Engineering. Washington: IEEE, 2006. 71−71. [doi: 10.1109/ICDE.2006.179]

[17] Ester M, Kriegel HP, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of the 2nd Int'l Conf.on Knowledge Discovery and Data Mining. AAAI Press, 1996. 226−231. [doi: 10.5120/739-1038]

© 中国科学院软件研究所 http://www.jos.org.cn

[18] Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. of the IEEE, 1989,77(2): 257−286. [doi: 10.1109/5.18626]

[19] Zheng Y, Xie X, Ma WY. Geolife: A collaborative social networking service among user, location and trajectory. IEEE Data Engineering Bulletin, 2010,33(2):32−40.

[20] Qiao SJ, Tang CJ, Jin HD, Long T, Dai SC, Ku YC, Chau M. PutMode: Prediction of uncertain trajectories in moving objects databases. Applied Intelligence, 2010,33(3):370−386. [doi: 10.1007/s10489-009-0173-z]

附中文参考文献:

[3] 孟小峰,丁治明.移动数据管理:概念与技术.北京:清华大学出版社,2009.185−200.

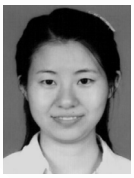
[4] 郭迟,刘经南,方媛,罗梦,崔竞松.位置大数据的价值提取与协同挖掘方法.软件学报,2014,25(4):713−730. http://www.jos.org.cn/1000-9825/4570.htm [doi: 10.13328/j.cnki.jos.004570]

[7] 乔少杰,韩楠,王超,祝峰,唐常杰.基于路网的移动对象动态双层索引结构.计算机学报,2014,37(9):1947−1958. [doi: 10.3724/SP.J.1016.2014.01947]

[15] 乔少杰,金琨,韩楠,唐常杰,格桑多吉,Gutierrez LA.一种基于高斯混合模型的轨迹预测算法.软件学报,2015,26(5):1048−1063. http://www.jos.org.cn/1000-9825/4796.htm [doi: 10.13328/j.cnki.jos.004796]

乔少杰(1981－),男,山东招远人,博士,副教授,CCF 高级会员,主要研究领域为移动对象数据库,轨迹数据挖掘.

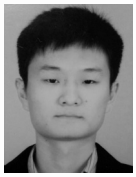
李天瑞(1969－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为智能信息处理.



韩楠(1984－),女,博士,工程师,主要研究领域为移动对象数据库.

高云君(1977－),男,博士,副教授,博士生导师,CCF 高级会员,主要研究领域为时空数据库.

元昌安(1964－),男,博士,教授,CCF 会员,主要研究领域为数据挖掘.



王晓腾(1991－),男,硕士生,主要研究领域为移动对象数据库,轨迹预测.

唐常杰(1946－),男,教授,博士生导师,CCF 杰出会员,主要研究领域为数据库,数据挖掘.

© 中国科学院软件研究所 http://www.jos.org.cn