

# 基于 CNN 和 LSTM 的移动对象目的地预测



李冰荣 皮德常 侯梦如

南京航空航天大学计算机科学与技术学院 南京 211106

(libingrongnuaa@163.com)

**摘要** 移动对象目的地预测是基于位置的服务的重要组成部分。该领域一直存在数据稀疏、长期依赖等难以解决的问题。为了有效解决这些问题,首先引入了一种基于最小描述长度策略(Minimum Description Length,MDL)的轨迹分段方法,以获得轨迹的最佳分段,提高轨迹之间的相似度,实现对轨迹的简化。随后将分段后的数据进行图像化处理和局部特征提取,并对轨迹目的地进行聚类,从而为轨迹数据增加标签。最后提出了一种基于卷积和长短期记忆循环单元的深度学习方法 CNN-LSTM,该算法先将局部图像数据和标签作为卷积神经网络(Convolutional Neural Network,CNN)模型的输入,通过空间特征的深度提取来保留有效信息,再利用长短期记忆网络(Long-Short Term Memory,LSTM)算法进行训练和目的地预测。在移动对象的真实轨迹数据集上进行了大量实验,结果表明,所提 CNN-LSTM 方法具有较强的学习能力,能更好地捕捉轨迹时空相关性。与现有的最新相关算法相比,该方法具有很高的目的地预测准确度。

**关键词:** 移动对象;目的地预测;轨迹;CNN;LSTM

中图法分类号 TP311

## Destination Prediction of Moving Objects Based on Convolutional Neural Networks and Long-Short Term Memory

LI Bing-rong, PI De-chang and HOU Meng-ru

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

**Abstract** Destination prediction of moving objects is an important part of location-based service. There are always some difficult problems in this field, such as sparse data and long-term dependence. In order to solve these problems effectively, firstly, a trajectory segmentation method based on the minimum description length strategy (MDL) is introduced, which can obtain the best trajectory segmentation, improve the similarity between tracks and realize the simplification of trajectories. Then, the segmented data are processed by image processing and local extraction, and the trajectory destination is clustered to add labels to the trajectory data. Finally, this paper proposes a deep learning framework CNN-LSTM based on convolution and long-short term memory. In this framework, local image data and labels firstly are taken as the input of the CNN model, and the effective information is preserved through the depth extraction of spatial features. Then, the LSTM algorithm is used for training and destination prediction. Extensive experiments are carried out on real trajectory dataset of moving objects. The results demonstrate that the CNN-LSTM method proposed in this paper has a strong learning ability and can better capture the spatiotemporal correlation of trajectories. In comparison to state-of-the-art and latest prediction algorithms, this method has high accuracy of destination prediction.

**Keywords** Moving objects, Destination prediction, Trajectory, CNN, LSTM

### 1 引言

随着无线定位、网络通信和数据挖掘技术的发展,利用 GPS 技术为出租车司机提供商业服务逐渐引起了人们的关注。基于位置的服务是一种通过移动设备和无线网络为用户提供地理位置的信息,其内容主要包含定位查询、路径规划和位置共享等功能。这些功能多集中于为用户提供当前位置的相关服务,所谓移动对象的目的地预测就是对用户未来

轨迹的终点位置进行预测,近年来国内外很多学者都在这方面做了大量的研究工作<sup>[1]</sup>,主要包含以下几个方面:

(1)基于位置的个性化推荐。用户的手持移动设备获取用户的历史轨迹数据,预测用户的下一个位置,进而针对用户进行个性化推荐。

(2)智能化交通。通过大量轨迹数据预测某一时段某一路段的交通状况,便于司机规划行驶路线,提高出租车的调度效率。

收稿日期:2020-02-05 返修日期:2020-06-19 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(U1433116)

This work was supported by the National Natural Science Foundation of China(U1433116).

通信作者:皮德常(dc.pi@nuaa.edu.cn)

(3) 犯罪分子追踪。移动用户位置预测可用于对犯罪嫌疑人进行定位追踪。公安机关利用智能手机可以便捷地获取犯罪嫌疑人的历史轨迹数据,分析挖掘出犯罪嫌疑人活动的热点区域、出行路线和犯罪行为等,从而预测出犯罪嫌疑人潜在的地理位置。

(4) 导航系统。用户的车载 GPS 可以记录其历史轨迹信息,挖掘其运动模式规律,预测其目的地,从而为其推荐最优路线或合适的停车场。

时空轨迹数据具有空间性、时间性、多维性、海量性、复杂性等特点。目前常用的移动对象目的地预测方法主要有两种类型:一种是挖掘历史轨迹频繁模式,然后用于预测;另一种是基于马尔可夫统计模型,对历史轨迹进行建模并预测。如果要查询的轨迹与历史轨迹中的某一部分相匹配,那么这部分轨迹的目的地很可能也是所查询轨迹的目的地。但这种方法存在一个很大的问题,即忽略了历史数据的稀疏性。数据稀疏是由于现实天气、道路、人为等各种因素导致移动对象的轨迹不完全一样,如果现有的历史轨迹远远不足以覆盖所有可能的查询轨迹,就会面临数据稀疏问题,进而造成预测的目的地位置不精确。当前解决数据稀疏性的方法有网格划分和相似度匹配,但它们也有一定的缺陷。网格划分方法会导致轨迹信息丢失,而相似度匹配方法在历史信息较少的情况下完全匹配率很低,无法用于轨迹预测。

针对以上问题,本文提出了一种基于卷积和长短期记忆的深度学习模型 CNN-LSTM。通过 CNN 模型进行空间特征的深度提取,以保留有效信息,然后利用 LSTM 算法进行训练和目的地预测。这样可以使模型具有时间、空间的特征表达能力,同时在挖掘移动对象运动规律的时具有较好的预测能力。本文主要贡献如下:

(1) 通过引入基于最小描述长度(MDL)的轨迹分段方法,将轨迹分段问题转化为求无向图的最短路径问题,从而求得轨迹的最佳分段,在一定程度上克服了轨迹预测时的数据稀疏问题。

(2) 将分段后的数据进行图像化处理和局部特征提取,从二维图像中获取移动对象的运动规律和空间模式。然后对轨迹目的地聚类,为轨迹图像增加标签,从而将轨迹目的地预测问题转化为轨迹分类问题。

(3) 采用 kaggle-ECML/PKDD 竞赛真实轨迹数据集<sup>[2]</sup>进行实验验证,使用预测准确率和误差来评价算法的性能。实验结果证明,CNN-LSTM 模型与当前现有的相关算法相比,具有较高的预测准确率。

本文第 2 节回顾了相关工作;第 3 节详细介绍了本文使用的轨迹分段方法、图像化处理和目的地聚类技术;第 4 节对 CNN-LSTM 的预测模型进行了具体介绍;第 5 节进行了实验并对结果进行了分析;最后总结全文并展望未来。

## 2 相关工作

随着移动传感器和智能手机设备的普及,人们越来越多地受益于各种位置服务。国内外学者已经针对移动对象轨迹预测展开了相关研究,并在此领域取得了一定的研究成果。轨迹预测模型主要划分为以下 3 类:

(1) 基于序列分析的预测方法。Morzy 提出了使用 Apriori 算法对轨迹进行关联挖掘<sup>[3]</sup>,并在后来的研究中利用改进的 PrefixSpan 算法对序列进行规律挖掘<sup>[4]</sup>。然而实验结果表明,由于数据每次更新都要重新挖掘频繁轨迹,这两种频繁轨迹挖掘算法的预测准确率均较低。Monreale 等<sup>[5]</sup>提出了 WhereNext 方法,即从历史数据中提取具有几种不同运动行为的运动模式,同时利用 T-pattern tree 查询最佳匹配轨迹,借此预测用户未来的位置。Ying 等<sup>[6]</sup>使用轨迹的地理位置、时间和语义信息预测移动对象下一刻的位置,该方法是通过挖掘同类用户的常见行为特性来进行预测的。

(2) 基于马尔可夫统计的预测方法进行预测。文献<sup>[7]</sup>采用简单的 Markov 模型进行位置预测。但该文仅考虑了当前位置对将来位置的影响,导致预测精确度较低。Mathew 等<sup>[8]</sup>使用隐马尔可夫模型计算移动轨迹序列中的隐状态来计算移动对象概率最高的下一个位置。文献<sup>[9]</sup>使用混合 Markov 模型对行人移动路径进行预测,减少了 2 阶 Markov 预测模型的空间代价,有效地对行人未来位置进行了预测。

(3) 目前的新兴方向主要是利用循环神经网络将层与层之间进行连接来处理序列数据,进而训练出有效模型。但循环神经网络不擅长学习长时间序列,不能解决数据之间的长期依赖问题。当查询轨迹预测目的地依赖的轨迹点数量增加,且轨迹预测的相关点离输出时刻太远时会出现长期依赖的问题。此外,由于循环神经网络的时间复杂度较高,导致预测准确率下降。Wu 等<sup>[10]</sup>提出一种基于 LSTM 的时空语义算法对移动对象进行下一位置的预测。Liu 等<sup>[11]</sup>提出 ST-RNN 算法,用移动对象历史轨迹数据训练 RNN 网络,预测用户在某一时刻的位置。

综上所述,基于序列分析的预测方法由于欠缺对移动对象内在的隐含信息的考虑,存在数据稀疏的问题。而基于马尔可夫统计的预测方法只依据前几个 GPS 点,忽略了历史数据的时效性。此外,基于单一的传统神经网络的方法不能准确捕获整个行为序列的时序性和局部关联性,导致预测的准确率偏低。针对以上方法的不足,为提高预测准确率,本文提出了基于 CNN 和 LSTM 双通道的融合算法来实现移动对象的目的地预测。实验表明,此方法具有非常优秀的预测效果,为移动用户高精度位置服务提供了可能。

## 3 轨迹数据预处理及相关定义

### 3.1 目的地预测的基本架构

为有效解决移动对象目的地预测领域中的数据稀疏、长期依赖等问题,本文提出了基于 CNN 和 LSTM 的目的地预测模型,其基本思路如图 1 所示。

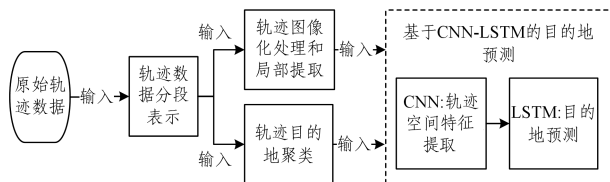


图 1 基于 CNN 和 LSTM 的目的地预测模型流程图

Fig. 1 Diagram of destination prediction based on CNN and LSTM

本文基于 MDL 方法提出了一种新颖的轨迹分段算法,即 IMDL(Improved Minimum Description Length)。此方法在很大程度上保存了原始轨迹特征信息,同时又能对每一条轨迹分段数据进行轨迹分割,解决了数据稀疏问题(详见 3.3 节)。然后,将分段后的数据进行图像化处理和局部特征提取,并对轨迹目的地进行聚类,从而为轨迹数据增加标签(详见 3.4 节)。

最后,先将局部图像数据和标签作为 CNN 模型的输入,通过空间特征的深度提取并保留有效信息,缓解了数据稀疏问题;然后利用 LSTM 进行训练和目的地预测,解决了数据的长期依赖问题(详见第 4 节)。

### 3.2 基本定义和问题描述

**定义 1(轨迹的序列表示)** 移动轨迹是指移动对象在地理空间中产生的 GPS 轨迹点的集合,通常由一系列按照时间顺序排列的点表示。给定欧氏空间,  $S = (p_1, p_2, \dots, p_n)$  为按照时间排序的离散轨迹点,其中  $p_i = (\lambda_i, \varphi_i, t_i)$  代表轨迹中的一个点,该点包含轨迹点的经度、纬度和时间。 $S_{1:i} = (p_1, p_2, \dots, p_i)$ ,  $i \leq n$  定义为车辆从开始到当前点的一部分轨迹。

**定义 2(预测轨迹序列)** 已知某一用户的轨迹序列  $S = (p_1, p_2, \dots, p_k)$ , 利用轨迹位置预测模型,得到预测轨迹序列  $S_p = \{S_{p_1}, S_{p_2}, \dots, S_{p_n}\}$ 。其中  $n > k$ ,  $S_{p_i}$  为第  $i$  个轨迹点。

#### (1)问题 1:数据稀疏问题

数据稀疏问题在轨迹预测中很常见。数据稀疏是指要查询的轨迹数据不足以覆盖所有可能的历史轨迹数据,也就无法满足轨迹预测的需求。

如果不解决数据稀疏问题,建立预测模型就会出现过拟合,对测试集数据的预测效果差,从而直接影响预测的准确率。因此,本文首先提出了一种改进的轨迹分段方法,在保留原始轨迹特征信息的情况下,减少相似轨迹之间的差异性,从而在一定程度上提高预测的准确率。

#### (2)问题 2:长期依赖问题

长期依赖是指当前系统的状态可能会受到很久以前的系统状态的影响。当查询轨迹预测目的地依赖的轨迹点数量增加,轨迹预测的相关点离输出时刻太远时会出现对历史数据的长期依赖。

如果长期依赖问题得不到解决,会导致预测的准确率降低。因此,本文采用 LSTM 网络进行建模,它在 RNN 的基础上增加了记忆模块,提高了 RNN 的鲁棒性,在很大程度上解决了对历史数据的长期依赖问题。

#### (3)问题 3:移动对象目的地预测问题

通过训练模型求解移动对象目的地位置很容易联想到直接对经度和纬度坐标进行预测。由于这种方式所需要的时间复杂度的成本较大,可取性不强,因此,本文采用一种新颖的预测方法,首先通过轨迹分段降低数据稀疏性;接着以轨迹目的地聚类的方式为轨迹图像增加标签,从而将轨迹目的地预测问题转化为轨迹分类问题;最后通过建立模型得到待预测轨迹目的地所在的区域类别。该方法不但降低了直接对经纬度坐标进行预测的难度,还具有很强的实用性,为实际应用提供了技术支撑。

### 3.3 轨迹分段

为了使轨迹更加简洁和方便处理,需要对轨迹进行分段。

MDL 被证明是行之有效的轨迹分段方法之一,因此本文提出一种改进型的 MDL 方法对轨迹进行简化。根据文献[12]的定义,线段之间的距离由 3 部分构成,即垂直距离、平行距离和角度距离,图 2 展示了 3 种线段距离的定义和物理含义。

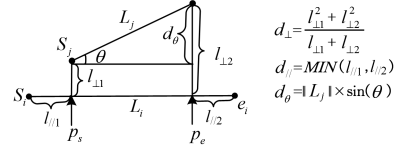


图 2 轨迹段之间的距离示意图

Fig. 2 Schematic diagram of distance between trajectory segments

根据上文所述,得出线段与线段之间的距离表示为:

$$\text{dist}(L_i, L_j) = \omega_{\perp} \cdot d_{\perp}(L_i, L_j) + \omega_{\parallel} \cdot d_{\parallel}(L_i, L_j) + \omega_{\theta} \cdot d_{\theta}(L_i, L_j) \quad (1)$$

其中,  $\omega_{\perp}$ ,  $\omega_{\parallel}$ ,  $\omega_{\theta}$  分别为垂直距离、平行距离和角度距离对应的权值,通常取 1。

最小描述长度的原理包含两个部分:  $L(H)$  和  $L(D|H)$ 。这里  $H$  是假设,  $D$  是数据。前者用来描述假设的长度,后者是在假设为  $H$  的情况下对数据  $D$  进行编码。当  $L(H) + L(D|H)$  的值最小时,  $H$  即为解释数据  $D$  的最佳假设。但该方法存在一个缺点,即它只能找到使  $L(H) + L(D|H)$  值较小的近似轨迹分段,而实际应用对轨迹的精确性和简洁性有一定的要求。

本文提出了一种改进型的最小描述长度方法,以满足对轨迹的简洁性和精确性的要求。这里采用  $mL(H) + nL(D|H)$  代替  $L(H) + L(D|H)$ , 即找到一个确定的分段,使得  $mL(H) + nL(D|H)$  的值最小,其中  $m, n \geq 0$ , 且  $m, n$  为实数,同时  $m, n$  的大小根据实际需求来调整,因此所得到的分段就是最佳分段。 $L(H)$  和  $L(D|H)$  的公式如下:

$$L(H) = \sum_{j=1}^{par_i-1} \log_2(\text{len}(p_{c_j}, p_{c_{j+1}}) + 1) \quad (2)$$

$$L(D|H) = \sum_{j=1}^{par_i-1} \sum_{k=c_j}^{c_{j+1}-1} \{\log_2(d_{\perp}(p_{c_j}, p_{c_{j+1}}, p_k, p_{k+1}) + 1) + \log_2(d_{\theta}(p_{c_j}, p_{c_{j+1}}, p_k, p_{k+1}) + 1)\} \quad (3)$$

接着将轨迹  $T_i$  中的所有点转化为一个无向完全图,则将轨迹分段问题转换为求无向完全图的最短路径问题,其中边  $p_i p_j$  的权重  $\omega(p_i p_j)$  由  $mL(H) + nL(D|H)$  计算得出。

图 3 给出了最佳分段问题转化为最短路径问题的示例。

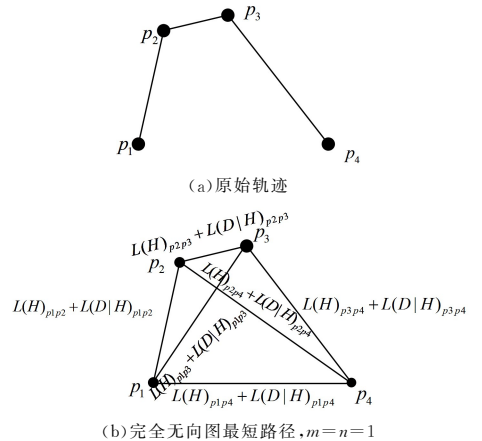


图 3 基于 MDL 的轨迹分段示例

Fig. 3 Example of trajectory segmentation based on MDL

图 3 是一条包含了 4 个点的轨迹,将轨迹中的任意 2 点进行连接,都可以求出对应的  $L(H)$  和  $L(D|H)$ 。由于可以求出每条边上的权重,因此可以基于 MDL 的原理计算出最佳分段,也就是该图中  $p_1$  到  $p_i$  的最短路径。算法 1 给出了轨迹分段算法 IMDL(Improved Minimum Description Length)。

#### 算法 1 轨迹分段算法 IMDL

输入:清洗后的轨迹数据  $T=(p_1, p_2, \dots, p_n)$

输出:分段后的轨迹集合 C

1. 将  $p_1$  点添加到集合 C;
2.  $startId=1, length=1, len=Len(T)$ ;
3.  $currentId=startId+length$ ;
4. WHILE  $currentId < len$  DO
5. 设置参数  $m, n$  的大小;
6. 计算代价函数  $\begin{cases} cost_{par} = IMDL_{par}(P_{startId}, P_{currentId}, m, n) \\ cost_{nopar} = IMDL_{nopar}(P_{startId}, P_{currentId}) \end{cases}$   
// 依据式(2)和式(3)
7. IF  $cost_{par} > cost_{nopar}$  THEN
8. 把  $p_{currentId-1}$  放入集合 C 中;
9.  $startId=currentId-1, length=1$ ;
10. ELSE IF  $cost_{par} \leq cost_{nopar}$  THEN
11.  $length=length+1$ ;
12. END IF
13.  $currentId=startId+length$ ;
14. END WHILE
15. 添加  $p_n$  到集合 C;
16. RETURN C.

IMDL 算法实现了对轨迹的分段。它首先初始化一个集合 C,将每条轨迹中第 1 个轨迹点加入集合中(第 1 行);然后根据实际对轨迹精确性和简洁性的要求来确定参数  $m$  和  $n$  的大小(第 4 行);通过计算代价函数来确定特征点(第 5—10 行),进而得到简化轨迹;最后输出轨迹分段后的数据集合 C。轨迹分段算法一方面提高了预测算法的运行效率,另一方面通过对长轨迹序列的特征提取,在保留原始有效信息的前提下,提高了预测的准确率。

#### 3.4 轨迹图像化处理和目的地聚类

由 3.3 节的 IMDL 轨迹分段算法可以得到原始轨迹分段后的数据集。目前大部分研究都是针对一维轨迹数据进行预处理,包括路网匹配和网格划分等。但对一维轨迹序列进行预处理只能获取移动对象在各个时间戳的位置信息,既不能得出轨迹的整体走向和目的地位置,也无法挖掘出详细的有效信息来预测位置。因此,在保留原始轨迹信息的前提下,需要进一步挖掘有效信息以解决数据稀疏问题。本节将对轨迹数据进行图像化处理,把一维轨迹序列转化为二维图像,即用像素矩阵来表示轨迹点,一方面保留了原始轨迹的有效信息,另一方面缓解了数据稀疏问题。

在图像处理领域,图像模式分为 3 种:灰度图像、黑白图像和彩色图像。本文选择黑白图片,像素值为 0 或 1。先提取移动对象的行驶轨迹并用像素坐标表示;然后完成路段内各轨迹点及其相对应的像素矩阵的构建;接着确定每一条轨迹的四方形区域,并尽可能地在该区域中划分小尺寸的网格,单个网格大小可设置为  $N \times N$ ,像素矩阵用  $M$  表示,每个网格对应一个像素点,轨迹每经过一个网格点,就将像素矩阵对

应位置的元素设置为 1,否则为 0;最终可以得到轨迹数据的像素矩阵  $M$ 。

在对未知轨迹进行历史轨迹匹配以确定所查询轨迹的目的地位置时会出现以下几种情况:

- (1)相似轨迹的目的地不一定相同。
- (2)相同目的地的两条轨迹的整体相似度不一定很高。
- (3)整条轨迹的某一部分与历史轨迹高度相似,目的地几乎相同,尤其是开始和结束时周围的轨迹。

因此,在对 CNN 模型进行输入时,本文选取的是局部图像,而非整段轨迹的图像。由于起点和终点附近的信息对目的地预测起着至关重要的作用,本文决定截取起点和终点附近不同大小的区域,截取比例分别为 20%,30%,40% 和 50%,然后将截取得到的不同大小的轨迹图像作为模型的输入数据,并对依次它们进行训练,最后得出所对应的预测误差(详见第 5 节)。

移动对象目的地预测的常用方法是直接对经度和纬度这一数值对进行预测,但本文并没有采取此种方法。因为模型的输入包含的信息量有限,如果直接预测经纬度坐标,时间复杂度和难度都势必大大增加,所以本文采用轨迹目的地的聚类的方式为轨迹图像增加标签,从而将轨迹目的地预测问题转化为轨迹分类问题,这不仅降低了预测复杂度,而且为位置服务的实际应用提供了可能。为保证本文研究成果与文献[13]有可比性,本文采用了同样的 mean-shift 聚类方法。该方法可以自动确定簇的个数,但划分的簇数会影响其预测效果。我们通过查阅很多相关文献发现,文献[14]采用了与本文相同的数据集进行轨迹聚类并做了大量的实验验证,因此,我们借鉴了其最佳划分簇数,将簇数设为 45。

## 4 基于 CNN 和 LSTM 的目的地预测算法

### 4.1 CNN 与 LSTM

卷积神经网络(CNN)模型是一种前馈神经网络,其基本结构由输入层(input)、卷积层(Conv)、池化层(Pool)、全连接层(Fc)和输出层(Output)交叉堆叠而成。其中卷积层由多个特征面组成;池化层紧接着卷积层,起到二次提取特征的作用,池化操作可以降低并防止过拟合。本文采用的是最大池化,因为它特别适用于分离非常稀疏的特征。随着层数的不断增加,提取的特征越来越抽象,该模型将这些抽象特征通过全连接层汇合,并通过输出层激活函数得出结果。

CNN 模型的结构如图 4 所示。该模型更擅长处理图像,即便对输入的图像进行一定程度的平移、缩放和旋转也不会影响处理效果。

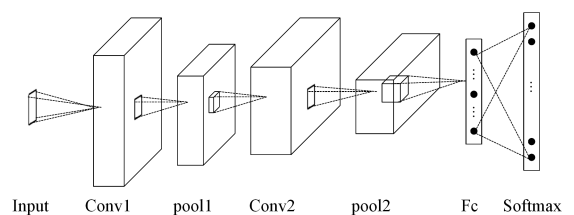


图 4 基于 CNN 的模型结构

Fig. 4 Model structure based on CNN

长短期记忆神经网络模型(LSTM)是建立在 RNN 基础

上的神经网络。由于 RNN 经过许多阶段传播后,梯度倾向于消失或爆炸,其所面临的挑战就是对历史数据的长期依赖问题。LSTM 模型是有效解决长期依赖问题的模型之一,其核心是加入记忆模块。

LSTM 使用记忆单元的状态来保存历史信息,每个记忆模块有 3 个“门”结构,分别为输入门( $i_t$ )、遗忘门( $f_t$ )和输出门( $o_t$ )。它们用于控制记忆单元,其精准的门结构会对细胞的信息进行增加或者删除。因此,利用门控单元对输入信息做筛选,则输出状态信息可表示为  $y(x) = \sigma(W_x + b)$ 。其中  $W$  表示权重矩阵, $b$  表示偏移向量。LSTM 模型的单元结构如图 5 所示。

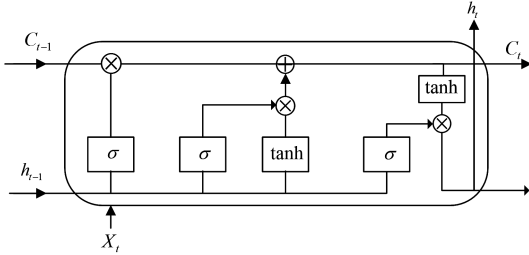


图 5 基于 LSTM 的模型结构图

Fig. 5 Model structure based on LSTM

设  $x_t$  为当前时刻  $t$  的输入时间序列,则各单元的状态可

以表示为以下公式。

输入单元:

$$i_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

门控单元:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

记忆单元:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (6)$$

输出单元:

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (7)$$

$$h_t = O_t \cdot \tanh(C_t)$$

## 4.2 CNN-LSTM 预测模型

CNN 与 LSTM 都是深度学习领域研究最热门、应用最广泛的模型。CNN 擅长图像的局部特征提取,LSTM 擅长捕获数据时序性,适合处理时间序列。由此可见,CNN 和 LSTM 在建模上具有一定的互补性。为提高模型的整体预测能力,使模型具有时间、空间的特征表达能力,本文提出了一种基于卷积和长短期记忆循环单元的深度学习框架 CNN-LSTM,其模型结构如图 6 所示。该模型将局部图像数据和标签作为 CNN 模型的输入,通过特征提取的方式来保留有效信息,再利用 LSTM 进行训练和目的地预测。

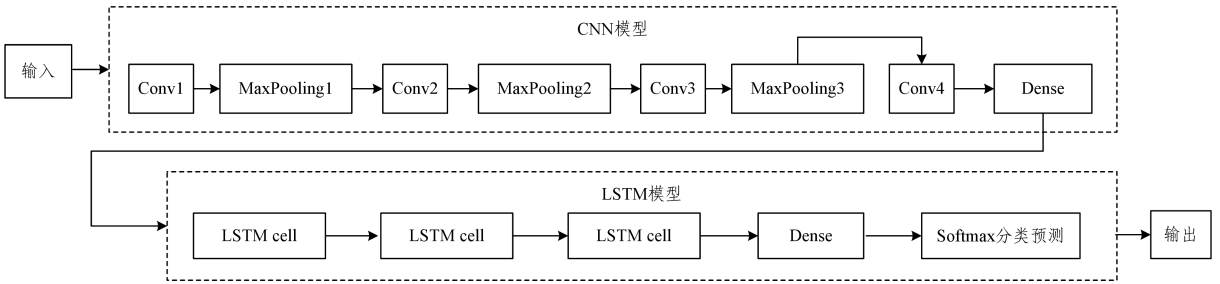


图 6 基于目的地预测的 CNN-LSTM 模型结构图

Fig. 6 CNN-LSTM structure based on destination prediction

图 6 中,CNN 负责局部轨迹的特征提取,LSTM 负责预测。CNN 结构由 4 层卷积层、最大池化操作和全连接层组成。其中卷积核大小分别为 16,32,64,128,池化层大小为 2,通过卷积池化操作得到  $2 \times 2 \times 128$  的三维向量数组。全连接层包含 50 个神经元,它的作用通常是在 CNN 的尾部进行重新拟合,以减少特征信息的损失。在 LSTM 模型部分,本文通过增加模型深度来提高预测能力,这里设置了 3 层 LSTM cell,其中隐藏层个数为 128。CNN-LSTM 模型的具体实现如算法 2 所示。

### 算法 2 CNN-LSTM 模型算法

输入:训练数据的局部图像的像素矩阵  $M$  与目的地聚类标签 Num\_label

输出:训练后的模型 BestModel

1. 初始化训练参数;
2. tensor=Cnn\_interface(M, parameters);  
//CNN 模型的训练
3. Model=init();
4. 数据序列化处理;
5. 构建多层 LSTM;

6. Model.add(LSTM());

7. FOR 第  $i$  步中的数据 DO;

8. Model.fit(seq\_len, Num\_label);

destination prediction CNN- LSTM 模型的训练

9. 计算损失函数;

10. 调整参数并优化模型;

11. END FOR

12. Model.evaluate();

destination prediction 训练模型评估

13. BestModel=save(Model);

14. RETURN BestModel.

算法 2 的基本思路如下。首先,初始化训练参数(第 1 行);接着,构建 CNN 模型,通过输入像素矩阵等训练参数,输出特征张量(第 2 行);然后,构建多层 LSTM 结构,实际训练深度设置为 3(第 3-6 行),其中对数据进行序列化处理(第 4 行),因为 LSTM 适用于时间序列数据;最后,将 CNN 模型训练提取的轨迹局部特征和目的地聚类后的标签 Num\_label 作为 LSTM 模型的输入,再将预处理后的数据放在模型中进行训练,并计算损失函数,通过一定次数的迭代,计算出

不同的迭代次数所对应的误差(第 7—11 行)。提取局部图像时,由于截取的区域大小不同,训练效果也不一样。因此,选取最好的训练模型保存并输出(第 12—14 行)。

### 算法 3 基于 CNN-LSTM 模型的目的地预测算法

输入:训练后的模型 BestModel 与待预测轨迹数据 Validate\_data

输出:待预测轨迹所在的区域类别集合 Range\_list

1. Test\_data=IMDL(Validate\_data);
2. 将待预测数据进行图像化处理;
3. FOR 第  $j$  个待预测数据 DO;
4. Batch\_x,y=get\_sequent\_data(Test\_data ,j); // 获取小批量的待预测数据
5. vali\_score=BestModel.add( Batch\_x,y); // 使用最好的训练模型进行预测
6. END FOR
7. Range\_list =Find\_position(vali\_score);  
// 获取待预测轨迹所在的区域类别集合
8. RETURN Range\_list.

算法 3 首先将待预测轨迹数据通过轨迹分段和图像化表示进行预处理(第 1—2 行);接着,对处理后的待预测数据进行划分(第 4 行),再利用算法 2 所训练的最好模型进行分批预测(第 5 行);然后,遍历每一条轨迹序列,通过目的地预测模型获取预测的准确率(第 3—6 行)。最后,输出待预测轨迹所在的区域类别集合 Range\_list(第 7—8 行)。

算法 3 并没有直接对目的地的经纬度坐标进行预测,而是把预测目的地限定在一定的区域范围内,把轨迹目的地预测问题转化为轨迹分类问题,其目的是最小化预测目的地与真实目的地之间的误差,这在文献[15]中已得到有效证明。算法 3 最后通过建立模型得到待预测轨迹目的地所在的区域类别。其不但具有非常高的预测准确率,而且提高了算法的实用性。

## 5 实验与分析

### 5.1 实验数据集与实验环境

本文采用了 kaggle-ECML/PKDD 竞赛上的真实轨迹数据集<sup>[22]</sup>。该数据集提供了葡萄牙波士顿图市 442 辆出租车一年的数据。本文从训练集中随机选取了 60963 条轨迹作为验证集,19770 条轨迹作为测试集,其余的作为训练集。

实验采用的硬件平台为 Intel Core i5-3230,内存 16 GB。实验软件为 Anaconda3 Jupyter Notebook,操作系统平台为 Windows 7,实验代码使用 python 编程实现。

### 5.2 评价方法

#### (1) 预测准确率

已知轨迹序列  $T=(p_1, p_2, \dots, p_n)$ , 预测轨迹序列定义为  $T_p=\{T_{p_1}, T_{p_2}, \dots, T_{p_n}\}$ , 则预测准确率的定义如式(8)所示:

$$Accuracy = \frac{\sum_{i=1}^n H(T_i, T_{p_i})}{|T_p|} \quad (8)$$

$$H(T_i, T_{p_i}) = \begin{cases} 1, & (d(T_i, T_{p_i}) < \sigma) \\ 0, & (d(T_i, T_{p_i}) > \sigma) \end{cases}$$

其中,  $T_{p_i}$  表示第  $i$  个样本的预测结果,  $T_i$  为真实结果,  $d(p, q)$  表示空间中  $p, q$  两点间的欧氏距离,  $\sigma$  表示距离阈值(如设

置为 1 km)。当  $d(T_i, T_{p_i}) < \sigma$  时,表示在阈值范围内的预测相对准确,所预测的目的地接近真实目的地,表示命中。

#### (2) 距离误差

本文采用 Haversine 距离来计算真实目的地与预测目的地之间的距离偏差,并将偏差作为度量预测模型的标准,其根据经度和纬度测量球面上两点的距离,具体公式如下:

$$\alpha = \sin^2\left(\frac{\lambda_1 - \lambda_2}{2}\right) + \cos(\lambda_1)\cos(\lambda_2)\sin^2\left(\frac{\varphi_1 - \varphi_2}{2}\right) \quad (9)$$

$$dis_{haversine} = 2R \arctan\left(\sqrt{\frac{\alpha}{1-\alpha}}\right)$$

其中,  $\lambda_1$  与  $\lambda_2$  是两个坐标点的纬度;  $\varphi_1$  与  $\varphi_2$  为两个坐标点的经度;  $dis_{haversine}$  是经过计算后两点间的距离;  $R$  为地球的半径,一般取值为 6371 km。

### 5.3 模型参数与测试结果

为了验证预测模型的有效性,本文采用预测准确率和距离误差来量化算法的性能。在对模型进行训练时,所使用的数据集没有取完整的轨迹长度,而是按照不同的划分比例截取起点和终点附近的轨迹信息作为输入。所截取的区域分别以轨迹起点和终点为圆心,按照轨迹长度的  $K\%$  为半径画圆,由此得到不同大小的区域图片作为模型的输入。其中验证集在训练过程中可以及时发现参数或者模型的问题,能够对模型能力进行初步评估,进而计算和设置合适的参数值。

本文选取按 20%, 30%, 40%, 50% 的比例截取的区域图片作为输入,得到了对应的训练误差和测试误差,并由此获取了不同的模型。本文提出的 CNN-LSTM 模型的训练参数配置如下。

(1) 为了保持训练过程相对稳定,本文采用滑动平均的方法更新参数。基础学习率  $learning\_rate\_base$  设置为 0.001, 衰减率  $learning\_rate\_decay$  取值为 0.99。为了对神经网络进行小批量梯度更新,在截取轨迹长度为 20% 的训练集样本中,  $batch\_size$  设置为 130,  $image\_width$  和  $image\_height$  的大小为  $40 \times 40$ ; 在截取轨迹长度为 30% 的训练集样本中,  $batch\_size$  设置为 150,  $image\_width$  和  $image\_height$  的大小为  $60 \times 60$ ; 在截取轨迹长度为 40% 的训练集样本中,  $batch\_size$  设置为 180,  $image\_width$  和  $image\_height$  的大小为  $80 \times 80$ ; 在截取轨迹长度为 50% 的训练集样本中,  $batch\_size$  设置为 200,  $image\_width$  和  $image\_height$  的大小为  $100 \times 100$ 。

图 7—图 10 分别展示了本文提出的预测模型算法 CNN-LSTM 在不同截取比例下的训练误差和测试误差。

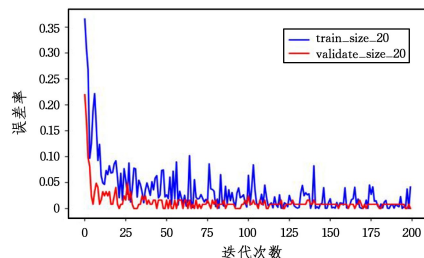


图 7 20% 的轨迹长度对应的训练误差和测试误差  
Fig. 7 Corresponding training error and test error of 20% of data length

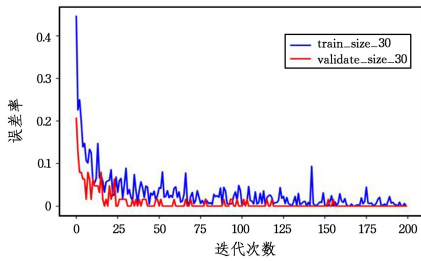


图 8 30%的轨迹长度对应的训练误差和测试误差

Fig. 8 Corresponding training error and test error of 30% of data length

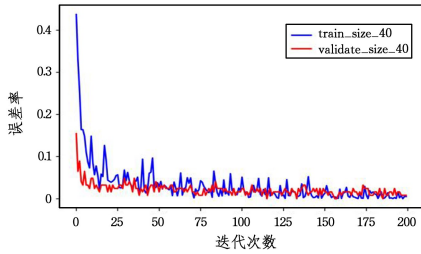


图 9 40%的轨迹长度对应的训练误差和测试误差

Fig. 9 Corresponding training error and test error of 40% of data length

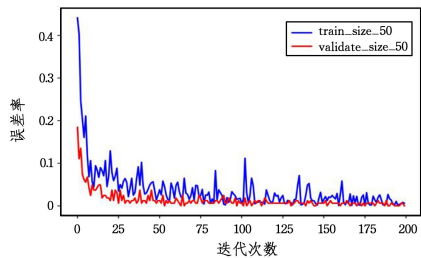


图 10 50%的轨迹长度对应的训练误差和测试误差

Fig. 10 Corresponding training error and test error of 50% of data length

从图 7—图 10 可以看出,在训练过程中,随着迭代次数的增加,误差逐渐减小,当迭代达到一定次数后,训练误差率趋于稳定,此时模型训练完毕。此外,可以看出,30%的截取比例比 20%,40%和 50%的截取比例的训练效果更好、更稳定,由此可见截取比例过大或过小都不好。

模型训练完成后,采用测试数据来验证模型。如果测试集的误差小于训练集的误差,则预测效果良好。由图 7—图 10 可知,测试集的误差都小于训练集的误差,避免了训练过程中的过拟合。因此,IMDL 的轨迹分段和图像化处理方法有效缓解了数据稀疏问题。

#### 5.4 算法比较与分析

为了验证 CNN-LSTM 预测模型的有效性,本文在 kaggle-ECML/PKDD 真实轨迹数据集上进行了大量的实验,并从预测准确度方面对 CNN-LSTM, LSTM, CNN, Subsyn 和 Baseline 5 种算法进行了实验比较。其中 LSTM 算法<sup>[16]</sup>擅长解决时序数据中的长期依赖问题,它利用 seq2seq 的模式,通过历史轨迹数据预测未来位置。CNN 算法<sup>[17]</sup>在图像识别领域具有出色的特征提取和特征抽象能力,能够分析轨迹行为的时空规律,预测目的地位置。文献<sup>[18]</sup>采用轨迹相似度,先

找出前  $N$  条最匹配的轨迹,然后根据搜索的这  $N$  条轨迹计算出终点的经度和纬度的均值,作为预测目的地的经纬度,这种算法被称作 Baseline 算法。Subsyn 算法<sup>[19]</sup>将历史轨迹序列划分成只包含两个相邻位置的子轨迹,通过连接子轨迹形成完整的轨迹来进行预测。图 11 分别给出了本文所提 CNN-LSTM 算法与现有的预测算法在实验数据上的预测准确率。

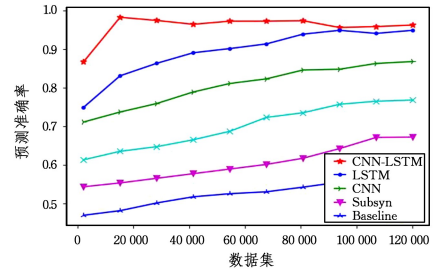


图 11 不同算法的预测准确率比较

Fig. 11 Comparison of prediction accuracy of different algorithms

纵向来, CNN-LSTM 的预测准确率明显高于 LSTM 算法、CNN 算法、Subsyn 算法以及 Baseline 算法。一方面是因为本文采用了改进的轨迹分段方法和图像化处理方法,缓解了轨迹数据稀疏的问题,另一方面 CNN 算法能捕获行为序列的局部相关性,提取了时序数据的潜在特征,为 LSTM 神经网络模型提供了大量的有效输入,使得分类效果更优。虽然 LSTM 算法也取得了不错的预测效果,但由于其不是将整条轨迹作为模型的输入,导致 LSTM 记住的场景并未在预测时发挥很大作用。而 CNN 的分类预测效果比 LSTM 差,表明轨迹数据的序列性比局部关联性更能体现用户的行为模式。Baseline 算法整体的预测准确率都不太高,可能是受到轨迹数据稀疏性的影响较大。Subsyn 算法依赖于轨迹预处理,因此当历史轨迹空间过大,存在异常轨迹时,预测效果降低。综上, CNN-LSTM 模型优于目前最新的相关预测算法。

**结束语** 移动对象的目的地预测是当前研究的热点,为各类基于位置的服务提供了实用价值。但该领域一直存在轨迹数据稀疏和对历史数据的长期依赖问题。部分学者尝试用网格划分和相似度匹配方法来解决历史数据稀疏问题,但它们都无法有效利用历史信息。近几年,虽然深度学习的神经网络模型成为众多学者研究的热点,但传统的神经网络方法存在对历史数据的长期依赖问题,无法记忆长轨迹序列。

针对以上问题,本文首先研究了轨迹分段方法,消除了数据稀疏问题对预测精度造成的不良影响;随后,将分段后的数据进行图像化处理和局部特征提取,并对轨迹目的地进行聚类,从而为轨迹数据添加了标签;最后,在充分利用历史数据的情况下,提出了一种融合卷积和长短期记忆循环单元的深度学习框架 CNN-LSTM 来进行目的地预测。本文在真实数据集上进行了大量的对比实验,结果表明,本文提出的双通道 CNN-LSTM 算法预测准确度最高。同时,我们在实验过程中发现一个有趣的问题,随着数据集规模的不断增大,所提 CNN-LSTM 算法的预测准确率呈小幅下降趋势。经过分析,本文认为可能是对历史数据的预处理方法不当而导致该模型的训练效果不佳,这也是一个值得探究的新问题,需要进一步的优化。

在未来的工作中,我们将针对该融合模型进行进一步研究,细化时间属性,降低算法的时间开销。此外,研究轨迹数据的时空特性,将天气、时间(工作日和非工作日)和移动对象的速度等因素考虑在内,为基于位置的服务提供技术支持。

### 参 考 文 献

- [1] SHIH D H, SHIH M H, YEN D C, et al. Personal mobility pattern mining and anomaly detection in the GPS era[J]. *American Cartographer*, 2016, 43(1): 55-67.
- [2] Kaggle. Kaggle competition[OL]. <https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i>, 2015.
- [3] MORZY M. Prediction of moving object location based on frequent trajectories[C]// *Proceedings of the 21st International Symposium on Computer and Information Sciences*. Berlin, Heidelberg: Springer, 2006: 583-592.
- [4] PEI J, HAN J, MORTAZAVI-ASL B, et al. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth [C]// *Proceedings 17th International Conference on Data Engineering*. Heidelberg, Germany, IEEE, 2001: 215-224.
- [5] MONREALE A, PINELLI F, TRASARTI R, et al. Wherenext: a location predictor on trajectory pattern mining[C]// *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2009: 637-646.
- [6] YING J J C, LEE W C, WENG T C, et al. Semantic trajectory mining for location prediction[C]// *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2011: 34-43.
- [7] CHEN M, LIU Y, YU X. Nlpm: A next location predictor with markov modeling[C]// *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, Cham, 2014: 186-197.
- [8] MATHEW W, RAPOSO R, MARTINS B. Predicting future locations with hidden Markov models[C]// *Proceedings of the ACM Conference on Ubiquitous Computing*. New York: ACM, 2012: 911-918.
- [9] ASAHARA A, MARUYAMA K, SATO A, et al. Pedestrian-movement prediction based on mixed Markov-chain model[C]// *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2011: 25-33.
- [10] WU F, FU K, WANG Y, et al. A spatial-temporal-semantic neural network algorithm for location prediction on moving objects [J]. *Algorithms*, 2017, 10(2): 37-62.
- [11] LIU Q, WU S, WANG L, et al. Predicting the next location: a recurrent model with spatial and temporal contexts[C]// *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. Phoenix, Menlo Park: AAAI, 2016: 194-200.
- [12] LEE J G, HAN J, WHANG K Y. Trajectory Clustering: A Partition-and-Group Framework [C]// *proceedings of the ACM SIGMOD International Conference on Management of Data*. 2007: 593-604.
- [13] BREBISSEON A D, SIMONE, AUVOLAT A, et al. Artificial neural networks applied to taxi destination prediction[C]// *Proceedings of the 2015th International Conference on ECML PKDD Discovery Challenge*. 2015: 40-51.
- [14] ESSE P C, GUILLOUET B, LOUBES J M, et al. Destination Prediction by Trajectory Distribution Based Model [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2018, 19(8): 2470-2481.
- [15] JIANG J, ZHANG H F, PI D C, et al. Destination Prediction of Moving Objects Based on Convolutional Neural Networks[J]. *Journal of Chinese Computer Systems*, 2019, 40(12): 2519-2525.
- [16] PEI Z, QI X, ZHANG Y, et al. Human trajectory prediction in crowded scene using social-affinity Long Short-Term Memory [J]. *Pattern Recognition*, 2019, 93: 273-282.
- [17] LYU J, LI Q, SUN Q, et al. T-CONV: a convolutional neural network for multi-scale taxi trajectory prediction[C]// *2018 IEEE international conference on big data and smart computing (bigcomp)*. IEEE, 2018: 82-89.
- [18] WANG L, ZHONG Y, MA W. GPS-data-driven dynamic destination prediction for on-demand one-way carsharing system[J]. *IET Intelligent Transport Systems*, 2018, 12(10): 1291-1299.
- [19] XUE A Y, QI J, XIE X, et al. Solving the data sparsity problem in destination prediction[J]. *The VLDB Journal*, 2015, 24(2): 219-243.



**LI Bing-rong**, born in 1996, M.S candidate. Her main research interests include location prediction of moving object and data mining, etc.



**PI De-chang**, born in 1971, professor, Ph.D supervisor, is a senior member of China Computer Federation. His main research interests include data mining, big data management and analysis.