计算机科学与探索
Journal of Frontiers of Computer Science and Technology
1673-9418/2019/13(01)-0023-12
doi: 10.3778/j.issn.1673-9418.1801029

基于LSTM的移动对象位置预测算法*

高　雅+,江国华,秦小麟,王钟毓
南京航空航天大学 计算机科学与技术学院,南京 210016
+通讯作者 E-mail: gaoya@nuaa.edu.cn

摘　要:移动对象位置预测是基于位置服务的重要组成部分。现有的移动对象位置预测算法有基于马尔可夫链的算法、基于隐马尔可夫模型的算法、基于神经网络的算法等,然而这些算法都无法解决移动对象轨迹数据中位置过多带来的维数灾难问题。为了解决这一问题,提出了位置分布式表示模型(location distributed representation model,LDRM)。该模型将难以处理的表示位置的高维one-hot向量降维成包含移动对象运动模式的低维位置嵌入向量。随后,将该模型与基于长短期记忆网络(long short-term memory,LSTM)的位置预测算法结合为LDRM-LSTM移动对象位置预测算法。真实数据集上的实验表明,与现有算法相比LDRM-LSTM算法在预测准确性上有较大的提升。

关键词:位置预测;降维;移动对象;长短期记忆网络(LSTM)
文献标志码:A　**中图分类号:**TP311

高雅, 江国华, 秦小麟, 等. 基于LSTM的移动对象位置预测算法[J]. 计算机科学与探索, 2019, 13(1): 23-34.
GAO Y, JIANG G H, QIN X L, et al. Location prediction algorithm of moving object based on LSTM[J]. Journal of Frontiers of Computer Science and Technology, 2019, 13(1): 23-34.

Location Prediction Algorithm of Moving Object Based on LSTM*

GAO Ya+, JIANG Guohua, QIN Xiaolin, WANG Zhongyu
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Abstract: Location prediction of moving object is an important part in location based service. Existing location prediction algorithms of moving object include Markov chain, hidden Markov model, neural network, etc. However, existing algorithms cannot solve the problem of dimensionality disaster caused by too many positions of the moving object's trajectory data. In order to overcome this problem, this paper proposes a location distributed representation model (LDRM). The model reduces the dimension of one-hot vector which represents each location to a low dimension location embedding vector which concludes the moving object's moving pattern. After that, LDRM is

* The National Natural Science Foundation of China under Grant Nos. 61373015, 61300052, 61402225 (国家自然科学基金); the Project Funded by the State Grid Corporation of China (国家电网公司科技资助项目).
Received 2018-01-15, Accepted 2018-05-08.
CNKI网络出版: 2018-05-21, http://kns.cnki.net/KCMS/detail/11.5602.TP.20180515.1458.002.html

combined with location prediction algorithm based on long short-term memory (LSTM) neural network to get an overall algorithm called LDRM-LSTM. Experiment results on real dataset show that, there has been a major improvement of the LDRM-LSTM algorithm compared with the existing ones, in terms of prediction accuracy.
Key words: location prediction; dimension reduction; moving object; long short-term memory (LSTM)

1 引言

随着以全球定位系统(global position system, GPS)为代表的定位技术的日趋成熟和普及,通过移动定位设备获取到的移动对象轨迹数据越来越多。位置信息是用户最为重要的上下文信息之一,可为各类服务和应用提供重要支持,如智能交通系统(intelligent transport system,ITS)、智能导航系统、路线规划、推荐系统[1]、公共设施规划[2]和公共安全[3]等。大规模可用性极高的数据和基于位置服务(location based service,LBS)的应用的发展,使移动对象的轨迹信息逐渐成为研究热点。其中,位置预测是基于位置的服务的重要组成部分,提出有效方法以分析和预测移动对象位置具有重要意义。例如,在导航系统中,通过用户的车载GPS记录其历史轨迹信息,挖掘其运动模式并预测用户的目的地,推荐不拥堵的最优路线或尚有空位的停车场。在交通系统中,对城市车辆轨迹的位置预测有助于了解市民的出行规律,提前预知某一时段某一路线的交通状况。在基于位置的信息投递中,通过用户的手持移动设备获取用户的历史签到数据,分析并预测用户下一步的位置,手机应用中的广告推送功能可用此对其进行相关的广告投放。

移动对象位置预测的方法的实现步骤一般为:首先,抽象化真实数据,将地理位置信息和时间信息以易于处理的方式表示;然后,对抽象化后的轨迹数据建模,进一步挖掘其运动模式[4];最后,对当前用户的输入轨迹,根据习得的运动模式,预测该用户下一步最可能到达的位置。

基于移动对象轨迹数据的位置预测问题得到了广泛的研究,其中,应用较为广泛的方法有:马尔可夫链(Markov chain,MC)模型、轨迹频繁模式挖掘和循环神经网络(recurrent neural network,RNN)等。马尔可夫模型通过计算用户位置之间的转换概率建立转移矩阵,以推测下一步最可能到达的位置[5-6]。轨迹模式挖掘通过挖掘频繁模式找出典型运动模式[7]。循环神经网络通过隐藏层神经元的连接性处理序列数据以学习运动模型[8]。

虽然上述方法已经在一些应用场景中得到了很好的效果,但它们针对的都是短时间轨迹序列,没有考虑到过长的历史信息会带来的维数灾难,不能解决长序列轨迹的位置预测问题。用户的轨迹数据会随时间的增长而变多,GPS的采样周期短,收集到的数据多且连续,在学习预测模型前,应对位置序列进行预处理,通过网格化和分布式表示的方法,降低位置向量的维数。并且,传统的方法如轨迹频繁模式挖掘,普遍都是先将原始轨迹按密度聚类,将轨迹线段划分到不同的类簇中,以聚类序列表示轨迹。这虽然能在保证划分意义的前提下缩短序列长度,但基于密度的聚类方法不仅时间复杂度大,而且会忽略聚类区域之外的离散轨迹。

轨迹数据是有时间顺序的时空数据,而神经网络被证明在处理时序数据的数据挖掘和预测方面效果最佳。但历史信息具有时效性,考虑失效的历史信息会产生梯度消失或梯度爆炸问题[9],而忽略有效的历史信息会导致模型精确度下降。长短期记忆网络(long short-term memory,LSTM)[10]被证明是这一问题很好的解决方法。但LSTM也要求输入低维度的向量,否则循环神经网络隐藏层和输入层的神经元之间的全连接属性会导致严重的维数灾难,增大开销,降低模型的学习效率。

针对上述问题,为了对移动对象的时空数据进行更精确的建模从而进一步预测其未来位置,本文提出了位置分布式表示模型(location distributed representation model,LDRM),将难以处理的高维位置one-hot向量转化为包含移动对象运动模式的低维位置嵌入向量。在LDRM模型基础上,与基于LSTM的位

置预测算法结合为LDRM-LSTM移动位置预测算法。该算法在考虑移动对象行为具有相似性和时效性的基础上，对历史轨迹位置向量进行降维，从而有效提高了预测模型的效率和精确度。

本文的主要贡献如下：

（1）提出了一种位置分布式表示模型LDRM，将难以处理的高维位置one-hot向量转化为包含移动对象运动模式的低维位置嵌入向量。在不忽略移动对象运动规律的同时，有效降低了位置序列数据的维数，提高了训练模型的效率。

（2）考虑轨迹数据的时效性和移动对象行为的相似性，采用LSTM模型作为训练模型，学习一个全局的位置预测模型，提高了长序列轨迹数据的预测精度。

（3）实验所用的GPS轨迹数据为微软亚洲研究院提供的真实数据集Geolife，并使用一系列量化指标来评估预测得到的结果。实验结果证明，与现有的算法相比，提出的LDRM-LSTM算法的预测精度得到了明显提升。

2　相关工作

随着基于位置的服务的发展，移动对象位置预测的工作逐渐成为研究的热点，国内外研究者针对移动对象位置数据挖掘与预测已经展开了相关研究。在移动对象位置预测中，通常的方法是通过移动对象历史数据，预测移动对象的位置。

Koren等人[11]提出了基于矩阵因子分解的方法，通过分解包含地理位置信息的用户行为矩阵判断用户最可能访问的位置。Xiong等人[12]改进了这一算法，通过张量分解（tensor factorization，TF）的方式，同时考虑历史轨迹信息中的地理位置信息和时间信息。上述算法都基于对移动对象历史访问记录的挖掘，没有考虑当前移动对象位置及轨迹，得到的预测精度不高。

Monreale等人[13]提出了WhereNext方法，从历史轨迹数据中提取对象的轨迹模式，借此预测用户频繁访问的位置，同时利用T-pattern tree查询最佳匹配轨迹。Ying等人[14]提出了基于地理和轨迹的语义特征预测移动对象下一时刻位置信息的方法，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。Morzy[15]采用改进的Apriori算法来生成关联规则，并在后来的研究中利用改进的PrefixSpan算法[16]通过移动轨迹序列频繁项集来预测移动对象的位置[7]。然而，频繁轨迹挖掘算法效率低下，数据每次更新都要重新挖掘频繁轨迹，使预测效率降低。

马尔可夫链（MC）模型利用移动对象的历史轨迹预测位置，考虑轨迹的顺序特征，使用马尔可夫链描述移动对象在位置之间的转移概率，Rendle等人[17]提出一种用因式分解转移概率矩阵的方法扩展了MC模型，在时空数据的位置预测上取得了很好的效果。Mathew等人[18]使用隐马尔可夫模型（hidden Markov model，HMM）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。但多阶MC模型尤其是高阶MC模型，当数据增加时状态会呈爆炸式增长，转移概率矩阵规模的膨胀增加了预测的复杂度。

近年来，神经网络在数据挖掘与预测中的应用逐渐广泛。在神经网络模型中，RNN最适用于时序数据的预测。Liu等人[8]提出了ST-RNN（spatial temporal-RNN）算法，用移动对象历史时空数据训练RNN网络，预测用户在某个时间点的位置。Al-Molegi等人[19]改进了该算法，其提出的STF-RNN（spatial temporal featured-RNN）算法获得了更好的预测准确率。

然而传统的RNN无法处理大量历史数据的轨迹序列，过多的历史数据会导致参数训练时的梯度消失、梯度爆炸和历史信息损失等问题，改进的LSTM则可以解决这些问题。LSTM在很多领域已经获得了一定的成果，如机器翻译[20]、信息检索[21]和图像处理[22]等方面。Sutskever等人[20]提出了基于LSTM的神经网络语言模型，并运用于自然语言处理方面，实现序列到序列（sequence to sequence）框架用于机器翻译。Malhotra等人[23]将LSTM模型运用于时序数据异常检测领域，解决了无监督环境下采集到非传感器数据从而导致时间序列不可预测的问题，在对不可预测的时间序列和长时间序列的异常检测中，取得了很好的效果。

Wu等人[24]提出了一种基于LSTM的时空语义算法(spatial-temporal-semantic neural network algorithm, STS-LSTM)对移动对象进行位置预测。该算法提出一种特征提取的方法将路网轨迹转化为离散位置点并用LSTM模型进行位置预测,较传统算法精确度有了明显的提升,但该算法并没有考虑到长时间序列的数据压缩问题。

针对以上不足,本文提出一种基于LSTM的移动对象位置预测方法,利用LSTM模型解决长序列的位置预测问题,在学习预测模型前,先对历史数据进行了预处理,使轨迹的时空数据转化为具有上下文信息且维数较低的神经网络输入向量。摆脱了传统位置预测算法的弊端,有效提高了预测精度和效率。

3 算法描述

LDRM-LSTM算法整体框架如图1所示,算法由三部分组成,即用于将轨迹数据转化为位置序列的预处理部分,对高维位置向量进行降维处理的位置分布式表示模型LDRM部分和基于LSTM的根据历史位置序列学习运动模式的预测模型部分。下面将用3节对这三部分进行详细的说明。

本章所用的轨迹数据的符号表示及说明如表1所示。

Table 1 Description of symbol definition

表1 符号定义说明

符号	说明
L_i	第 i 个网格
N_{L_i}	第 i 个网格中位置点的个数
$S=\{L_1,L_2,\cdots,L_n\}$	位置序列
$T=\{\langle L_1,t_1\rangle,\langle L_2,t_2\rangle,\cdots,\langle L_n,t_n\rangle\}$	带时间属性的位置序列
$H=\{T_1,T_2,\cdots,T_n\}$	带时间属性的位置序列集合
$\boldsymbol{l}_i$	L_i 的one-hot向量
$\boldsymbol{e}_i$	L_i 的位置嵌入向量

3.1 轨迹数据预处理

移动对象轨迹数据通常由GPS采集。考虑轨迹数据的结构,经过采样生成的一条轨迹可以表示为一系列的位置点的集合。由于采集的点的序列,在时间空间上相对紧密,难以从中挖掘出其运动的模式。如图2中的两条轨迹P(实线轨迹)与P'(虚线轨迹),很难从中发现明显的普遍特征,那么对于大量的如此表现形式的轨迹,就更难挖掘出其中的模式进行预测。因此在移动对象位置预测之前,必须对原始轨迹数据进行预处理。

网格化是研究移动对象轨迹常用的方式,其核心思想是将移动对象所在平面划分为网格,将精确但冗余的经纬度信息转化为抽象的网格信息,以便对移动对象轨迹进行预测。如图2右图所示,通过网

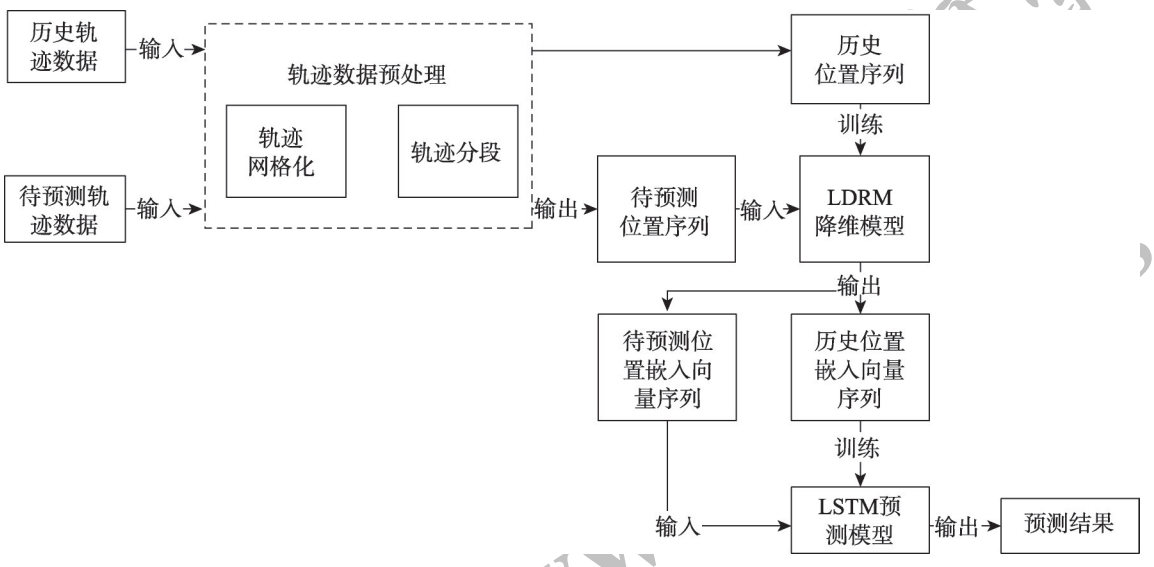

Fig.1 Overall framework of LDRM-LSTM

图1 LDRM-LSTM算法整体框架

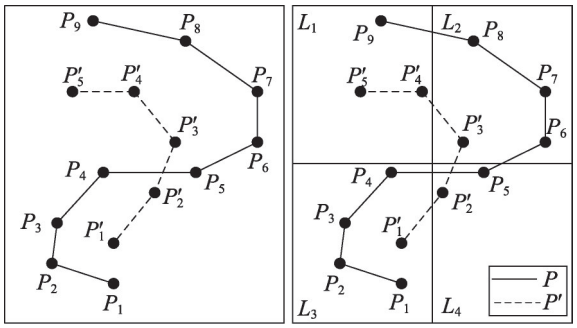

Fig.2　Trajectory data gridding

图2　轨迹数据网格化

格将复杂的轨迹 P 与 P' 转化为相同的轨迹 $L_1 \rightarrow L_2 \rightarrow L_4 \rightarrow L_3$，更能抽象地表示移动对象的运动过程。

采用Geohash编码对轨迹进行网格化。Geohash由Niemeyer提出，最初用于geohash.org服务中，目的是为地球上每一个位置提供一条短URL作为唯一标识[25]。Geohash编码的基本原理是将地球视为二维平面，把该平面沿经度和维度的方向递归二分为更小的子平面，使平面被网格化为子平面的集合，每个子平面在一定经纬度范围内拥有相同的Geohash编码。根据Geohash编码方案，其编码具有唯一性，即Geohash的每个单元网格在地球表面均有唯一空间区域与之对应，这种特性便于对轨迹数据进行无歧义的网格序列化。

使用Geohash编码将轨迹所在地区分为 n 个网格，第 i 个网格编码为 $L_i(1 \leqslant i \leqslant n)$，每个网格代表一个位置。由于GPS数据采集的时候可能有误差，有些网格只有极少数轨迹点存在其中，需要将其作为离群点剔除。因此，设计算轨迹经过编号为 L_i 的网格的轨迹点个数为 N_{L_i}，若 N_{L_i} 大于网格最小有效轨迹点数 δ，则将该网格编码 L_i 与来到该网格的时刻 t_i 构成的二元组加入带时间属性的位置序列 T，如式(1)所示。

$$T=\{<L_1,t_1>,<L_2,t_2>,\cdots,<L_N,t_N>|N_{L_i}>\delta\} \quad (1)$$

根据式(1)，将原始轨迹数据转化为带时间属性的位置序列 T。然而，GPS轨迹采集过程通常是连续很长一段时间，导致 T 过长，并且其中有些二元组之间的时间间隔过大，对位置预测有较大影响。因此，需要通过时间阈值将位置序列 T 分段。设第 k 段带时间属性的位置序列 $T_k=\{<L_k,t_k>,<L_{k+1},t_{k+1}>,\cdots,<L_{k+l},t_{k+l}>\}$，其中存在时间点 t_i，有 $t_{i+1}-t_i>\gamma$，则将位置序列 T_k 分为 T_a 和 T_b 两个子序列，用时间点 t_i 分割原序列，分割后的两个子序列 $T_a=\{<L_k,t_k>,<L_{k+1},t_{k+1}>,\cdots,<L_i,t_i>\}$，$T_b=\{<L_{i+1},t_{i+1}>,<L_{i+2},t_{i+2}>,\cdots,<L_{k+l},t_{k+l}>\}$，其中，$k<i<l$。

通过上述算法，将移动对象原始轨迹数据转化为 h 个位置序列，构成位置序列集合 $H=\{T_1,T_2,\cdots,T_h\}$。同时，构建 n 个移动对象位置组成的集合 $S=\{L_1,L_2,\cdots,L_n\}$。移动对象位置预测问题如定义1所示。

定义1（移动对象位置预测问题）设 h 个历史位置序列组成的历史轨迹集合 $H=\{T_1,T_2,\cdots,T_h\}$，当前轨迹为 $T'=\{<L_i,t_i>\},0 \leqslant i \leqslant t$，预测移动对象在 $t+1$ 所在的位置 L_{t+1}。

3.2　位置分布式表示模型LDRM

将移动对象轨迹数据转化为位置序列后，由于位置信息是离散的Geohash编码，无法直接进行训练，需要将编码转化为向量。通用的离散值转为向量的方法是one-hot编码，即对于 n 个位置的集合 $S=\{L_1,L_2,\cdots,L_n\}$，对于每一个位置 L_i，构建一个零向量，将其第 i 维赋值为“1”，即只有这一位有效，其他位均为“0”。如位置 L_1 的可表示为：$(1,0,0,0,\cdots)$，这个稀疏的向量即为该位置的one-hot编码。这种编码将离散的位置编码转化为向量，解决了模型无法处理离散数据的问题。但是这种编码也存在着问题：(1)在数据量大、不同位置多的情况下，容易受维数灾难的困扰；(2)不能很好地描述位置之间的关系，也不能体现出移动对象的运动模式。

针对one-hot存在的问题，采取分布式表示(distributed representation)的方法，提出了位置分布式表示模型LDRM，将包含位置信息的one-hot编码通过神经网络转化为低维度的含有位置上下文信息的位置嵌入向量(location embedding vector，LEV)，避免了由于位置过多带来的维数灾难问题。

移动对象的运动往往存在某种隐藏的规律，即某些移动对象序列会以较高概率出现。设一个长度为 N 的位置序列出现概率为 $P=(L_1,L_2,\cdots,L_N)$，在理

想状态下，即移动对象出现在每个位置的概率相互独立时，计算公式如式(2)所示。

$$P(L_1,L_2,\cdots,L_N)=\prod_{i=1}^{N}P(L_i) \tag{2}$$

但这样的假设太过理想化，因为移动对象当前位置与其之前位置是相关的。假设在一段位置序列中，移动对象的位置和前 t 个位置有关，如式(3)所示。

$$P(L_1,L_2,\cdots,L_N)=\prod_{i=1}^{N}P(L_i|L_{i-t},L_{i-t+1},\cdots,L_{i-1}) \tag{3}$$

LDRM模型在对位置one-hot向量降维的同时，从轨迹未知序列中挖掘位置之间的关系，使得 $P(L_i|L_{i-t},L_{i-t+1},\cdots,L_{i-1})$ 最大化，从而将序列中的运动模式隐含到位置嵌入向量中。

设移动对象位置序列共有 n 个位置，则每个位置构成一个 n 维的one-hot向量 $\boldsymbol{l}$，每个one-hot向量对应一个 m 维的位置嵌入向量。LDRM模型神经网络结构如图3所示。

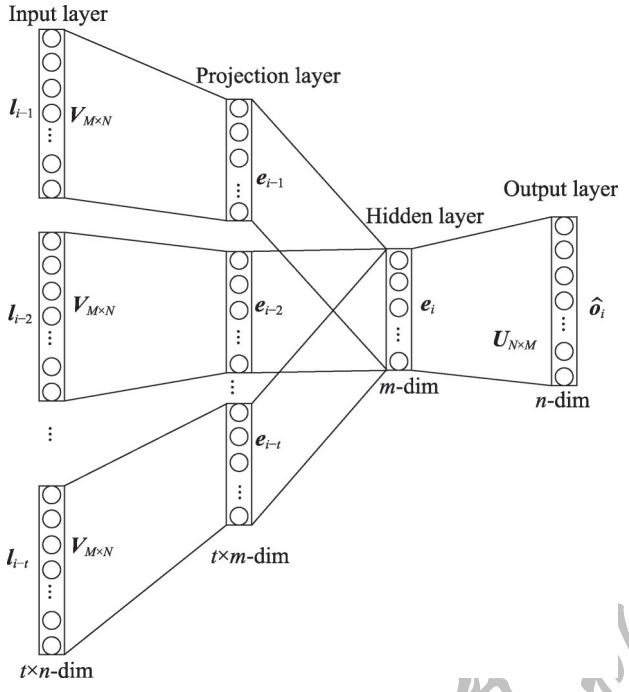

Fig.3 Structure of LDRM neural network

图3 LDRM神经网络结构

设当前位置编号为 i，对应的one-hot向量为 $\boldsymbol{l}_i$，和当前位置相关的位置为 $\{\boldsymbol{l}_{i-t},\boldsymbol{l}_{i-t+1},\cdots,\boldsymbol{l}_{i-1}\}$。假设当前存在输入矩阵 $\boldsymbol{V}\in\mathbb{R}^{m\times n}$，根据式(4)可计算投影层每个位置嵌入向量 $\{\boldsymbol{e}_{i-t},\boldsymbol{e}_{i-t+1},\cdots,\boldsymbol{e}_{i-1}\}$。

$$\boldsymbol{e}_r=\boldsymbol{V}\boldsymbol{l}_r, i-t\leqslant r\leqslant i-1 \tag{4}$$

在移动对象位置序列中，不同位置的重要程度不同，比如热门景点、商场、学校、医院等对下一个位置的影响要明显高于一般位置。故给每一个位置根据其在所有轨迹中出现的次数赋予一个权值 w_i，计算方式如式(5)所示。

$$w_i=\varepsilon\frac{count(l_i)}{\sum_{j=1}^{n}count(l_j)} \tag{5}$$

将前 i 个位置的嵌入向量通过式(6)计算出当前位置的嵌入向量 $\boldsymbol{e}_i$。

$$\boldsymbol{e}_i=\frac{\sum_{j=i-t}^{i-1}w_j\boldsymbol{e}_j}{t} \tag{6}$$

设输出矩阵 $\boldsymbol{U}\in\mathbb{R}^{n\times m}$，当前位置的嵌入向量经过输出矩阵的变换成为 n 维输出向量。输出层将嵌入向量转化为输出向量的计算方式如式(7)所示。

$$\hat{\boldsymbol{o}}_i=\text{soft max}(\boldsymbol{U}\boldsymbol{e}_i) \tag{7}$$

$\hat{\boldsymbol{o}}_i$ 为 n 维输出向量，理想状态下的输出向量 $\boldsymbol{o}_i$ 应该和当前位置的one-hot向量相等，即 $\boldsymbol{o}_i=\boldsymbol{l}_i$。在这种情况下，该模型的损失函数交叉熵(cross entropy)的计算方式如式(8)所示。

$$loss(\hat{\boldsymbol{o}}_i,\boldsymbol{o}_i)=-\sum_{j=1}^{n}\boldsymbol{o}_{ij}\ln(\hat{\boldsymbol{o}}_{ij}) \tag{8}$$

由于模型输出为离散的one-hot向量，若直接使用交叉熵作为目标函数，会造成模型泛化性严重下降。因此，定义该模型的目标函数如式(9)所示。其中，$\boldsymbol{u}_i$ 为输出矩阵 $\boldsymbol{U}$ 的第 i 行，表示 $\boldsymbol{l}_i$ 的输出向量。

$$\begin{aligned}&\text{min imize } J=-\ln P(\boldsymbol{l}_i|\boldsymbol{l}_{i-t},\boldsymbol{l}_{i-t+1},\cdots,\boldsymbol{l}_{i-1})=\\&-\ln P(\boldsymbol{u}_i|\boldsymbol{e}_i)=-\ln\frac{\exp(\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{e}_i)}{\sum_{j=1}^{m}\exp(\boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{e}_i)}\end{aligned} \tag{9}$$

使用梯度下降的方法，计算 $\boldsymbol{u}_i$ 和 $\boldsymbol{e}_i$，即可得到矩阵 $\boldsymbol{V}$ 和 $\boldsymbol{U}$。通过公式 $\boldsymbol{e}=\boldsymbol{V}\boldsymbol{l}$，即可计算每个one-hot向量对应的位置嵌入向量。位置嵌入向量的总体计算流程如算法1所示。

算法1 LDRM算法

输入：每个位置one-hot向量集合 S_{onehot}，轨迹序列集合 $S_{\text{trajectory}}$，相关位置数 t。

输出：每个位置对应的位置嵌入向量构成的集合 S_{LEV}。

1. *Trainingset*=[]; //构建训练数据集及其标签，初始值为空

2. For each l_i in S_{onehot} //遍历one-hot向量集合

3. For each T_j in $S_{\text{trajectory}}$ //遍历轨迹序列集合

4. If l_i in T_j and the order of l_i is k //如果当前位置在当前轨迹序列中出现，且是当前轨迹序列中第 k 个位置

5. Then $data=\{l_{i-t},l_{i-t+1},\cdots,l_{i-1}\}$，$label=l_i$ //当前位置的前 t 个位置为训练数据，当前位置为标签

6. add {*data*,*label*} into *Trainingset* //存储训练数据与标签

7. While $epoch<epochround$ //当训练未完成时

8. For each {*data*, *lable*} in *Trainingset* //遍历构建的数据集

9. Input {*data*, *lable*} into Nerual Network//将数据与标签放入神经网络

10. Calculate function J //计算目标函数

11. Backpropagation parameter U and V //反向传播调整参数 U 和 V

12. For each l_i in S_{onehot}

13. $\boldsymbol{e}_i=\boldsymbol{U}\cdot\boldsymbol{l}_i$ //对于每一个one-hot向量，计算对应的位置嵌入向量

14. Add $\boldsymbol{e}_i$ to S_{LEV} //存储计算结果

算法1计算了每个位置one-hot向量构成的位置嵌入向量。计算过程分为三个主要部分：首先通过遍历位置one-hot向量集合和轨迹序列集合构建训练集；然后使用该训练集训练LDRM神经网络参数 U 和 V；最后通过该参数计算每个位置的位置嵌入向量。

3.3　基于LSTM的位置预测算法

LSTM模型是一种RNN的变型，由记忆单元（memory cell）和多个调节门（gate）组成。LSTM使用记忆单元的状态（state）来保存历史信息，使用输入门（input gate）、输出门（output gate）和遗忘门（forget gate）来控制记忆单元。利用调节门来选择信息，可表示为 $y(x)=\sigma(\boldsymbol{W}x+\boldsymbol{b})$，其中 $\boldsymbol{W}$ 表示权重矩阵，$\boldsymbol{b}$ 表示偏移向量。由于传统的RNN展开后相当于多层的前馈神经网络，层数对应于历史数据的数量，过多的历史数据会导致参数训练时的梯度消失、梯度爆炸和历史信息损失等问题，在处理大量历史数据的预测问题上，LSTM被证明比RNN具有更好的表现。

LSTM的单元结构如图4所示，设 x_t、h_t 为 t 时刻的输入和输出数据，i、f 和 o 分别为输入层、遗忘层和输出层，C_t 为 t 时刻记忆单元的状态值。LSTM单元的更新可分为如下几个步骤。

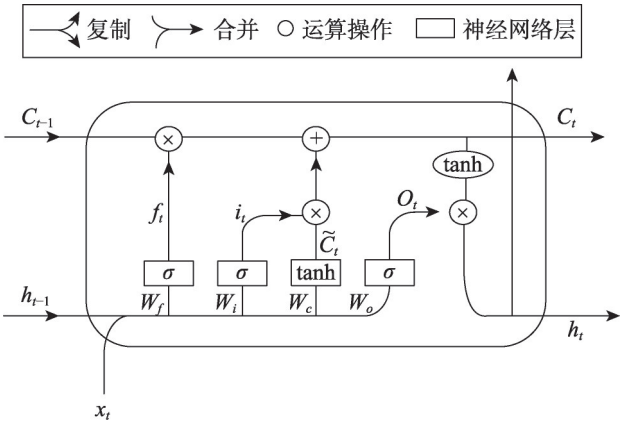

Fig.4　Structure of LSTM network

图4　LSTM单元结构图

（1）决定从上一状态中丢弃什么信息。遗忘门是用于控制历史信息对当前记忆单元状态值的影响的，因此通过遗忘门来计算 f_t，如式（10）所示。其中 $\boldsymbol{W}_f$ 表示遗忘门的权重矩阵，$\boldsymbol{b}_f$ 表示遗忘门的偏移向量。σ 是logistic sigmoid激活函数，取值在（0,1）之间，1表示“全部保留”，0表示“全部丢弃”。

$$f_t=\sigma(\boldsymbol{W}_f\cdot[h_{t-1},x_t]+b_f) \tag{10}$$

（2）决定什么信息被存入当前状态中。输入门是用于控制当前输入对记忆单元状态值的影响的，因此通过输入门来计算 i_t，如式（11）所示。按照传统的RNN公式计算当前时刻的候选记忆单元值 $\tilde{C}_t$，其中函数tanh的取值范围在（−1,1）之间。

$$\begin{cases} i_t=\sigma(\boldsymbol{W}_i\cdot[h_{t-1},x_t]+b_i) \\ \tilde{C}_t=\tanh(\boldsymbol{W}_c\cdot[h_{t-1},x_t]+b_c) \end{cases} \tag{11}$$

（3）决定好丢弃和更新什么信息后，执行记忆单元的更新，计算当前时刻记忆单元的状态值 C_t，如式（12）所示，$\odot$ 是点乘运算。

$$C_t=f_t\odot C_{t-1}+i_t\odot\tilde{C}_t \tag{12}$$

（4）确定输出值。输出门用于控制记忆单元状态值的输出，用sigmoid函数计算决定输出的部分，如式（13）所示。将通过tanh函数处理的记忆单元状态

与结果 o_t 相乘，得到LSTM单元在 t 时刻的输出 h_t。

$$\begin{cases} o_t = \sigma(\boldsymbol{W}_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t = o_t \odot \tanh(C_t) \end{cases} \tag{13}$$

经过LSTM根据训练数据调整神经元之间的权重的学习过程[26]，得到一个全局的位置预测模型。通过上述门控机制，LSTM解决了RNN处理长序列容易发生的梯度消失、梯度爆炸和历史信息损失等问题，在移动对象位置预测方面有更好的表现。

使用LDRM模型对位置序列进行降维后，难以处理的高维one-hot向量被转化为低维度的含上下文信息的位置嵌入向量，从而使得基于LSTM的位置预测算法可以更好地处理移动对象位置预测问题。

设降维后的位置嵌入向量集合为 $S_{\text{LEV}} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_n\}$，则移动对象位置序列可以转化为由嵌入向量组成的序列。但放入LSTM模型中的每个序列的长度必须相同，因此采取滑动窗口的方式，将所有的轨迹转化为可训练的固定长度的输入与输出样本集，如图5所示。

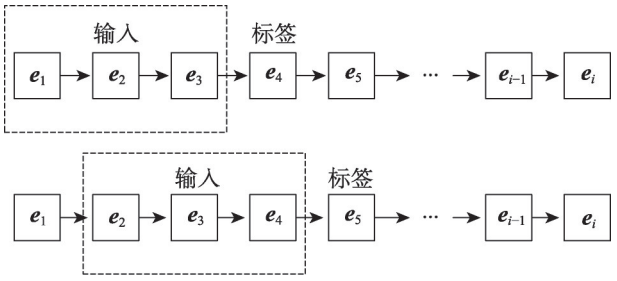

Fig.5 Generate sample set using sliding window method

图5 滑动窗口法生成样本集

将经过LDRM模型降维的历史位置嵌入向量作为LSTM模型的输入，训练出针对移动对象位置预测问题的LSTM模型。将待预测位置嵌入向量序列输入模型得到输出向量 $\boldsymbol{e}_{\text{output}}$，将 $\boldsymbol{e}_{\text{output}}$ 与 $\boldsymbol{E}$ 中所有位置嵌入向量计算欧式距离，与 $\boldsymbol{e}_{\text{output}}$ 距离最小的位置嵌入向量即为预测结果 $\boldsymbol{e}_{\text{result}}$。模型的位置预测结果即为 $\boldsymbol{e}_{\text{result}}$ 对应的位置one-hot向量 $\boldsymbol{l}_{\text{result}}$。

算法2 LSTM位置预测算法

输入：位置嵌入向量集合 S_{LEV}，轨迹序列集合 $S_{\text{trajectory}}$，待预测位置序列 T_{test}。

输出：预测结果的one-hot向量 $\boldsymbol{l}_{\text{result}}$。

1. *Trainingset*=[]; //构建训练数据集及其标签，初始值为空

2. For each T_i in $S_{\text{trajectory}}$

3. For slide window $T_s = \{l_a, l_{a+1}, \cdots, l_b\}$ in T_i //通过滑动窗口方式构建子序列

4. Extract $data = \{e_a, e_{a+1}, \cdots, e_{b-1}\}$, $label = e_b$ from T_s //将子序列中one-hot向量转化为位置嵌入向量，并构建训练数据与标签

5. Add {*data*, *label*} to *Trainingset* //保存训练数据与标签

6. While $epoch < epochround$ //当训练未完成时

7. For each {*data*, *lable*} in *Trainingset*

8. Input {*data*, *lable*} into LSTM neural network //将训练数据与标签输入LSTM网络

9. Calculate bias function of LSTM //计算误差函数

10. Backpropagation parameters //反向传播调整网络参数

11. Input T_{test} into LSTM Network get result embedding vector $\boldsymbol{e}_{\text{output}}$ //将测试数据输入网络得到输出的位置嵌入向量

12. For each $\boldsymbol{e}_i$ in S_{LEV}

13. Find $\boldsymbol{e}_{\text{result}} = \boldsymbol{e}_i$, for which distance of $\boldsymbol{e}_i$ and $\boldsymbol{e}_{\text{output}}$ is minimum //在位置嵌入向量集中寻找距离最近的向量

14. Output corresponding $\boldsymbol{l}_{\text{result}}$ of $\boldsymbol{e}_{\text{result}}$ //将距离最近的向量转化为one-hot向量并输出

LSTM位置预测算法的流程如算法2所示。首先通过滑动窗口将轨迹数据分割并转化为训练集。然后通过训练集训练LSTM神经网络，最后将待预测轨迹输入LSTM神经网络中，得到输出的位置嵌入向量，并筛选与该向量距离最近的位置嵌入向量，该向量对应的位置即为预测的位置。

由算法1与算法2可见，LDRM-LSTM算法采用了离线训练在线预测流程结构，即在预测前将模型参数训练完成，在预测时仅需要计算预测结果即可。这种方式的优点为将耗时操作在预测前完成，预测时的时间复杂度极低。本文中，LDRM降维模型和LSTM预测模型均由历史轨迹数据离线训练得出，在进行轨迹数据预测时只需将待预测轨迹数据进行预处理后输入模型即可得到预测结果，具有很强的实用性。

4　实验与分析

4.1　实验数据与环境

为了验证移动对象位置预测算法的预测准确性,使用微软亚洲研究院的Geolife project数据进行测试。该数据集包括182个用户5年间(从2007年4月到2012年8月)的运动轨迹数据。数据集覆盖中国、美国和欧洲的不同城市,大部分数据是在北京采集的。轨迹数据数量为18 670条数据,覆盖长达50 176 h的时间和1 292 951 km的距离。轨迹是通过带GPS功能的手机采集而来,每隔1到5 s采集一次位置数据。轨迹信息中包含经纬度、时间等信息。

实验操作系统为ubuntu14.04(64 bit),CPU为Intel Core i5-3470,内存为16 GB,显卡为NVIDIA GTX970,实验代码用python编程语言实现。

4.2　预测评价标准

定义2　绝对预测精度(absolute precision, AP)即预测结果和真实结果的差距。设共有 K 个测试样本,第 i 个样本的预测结果为 $\boldsymbol{pl}_i$,真实结果为 $\boldsymbol{l}_i$。绝对预测精度分数 $Score_{\mathrm{AP}}$ 如式(14)所示。

$$Score_{\mathrm{AP}}=\sum_{i=1}^{K}s_i/K \tag{14}$$

$$s_i=\begin{cases}1, \boldsymbol{pl}_i=\boldsymbol{l}_i\\0, \boldsymbol{pl}_i\neq\boldsymbol{l}_i\end{cases}$$

由于绝对预测精度的要求较为苛刻,无法对算法在预测不完全精确的情况下的预测结果优劣进行比较。因此实验同时使用相对预测精度来进行预测精度度量。

定义3　相对预测精度(relative precision, RP)即在预测不完全精确的情况下,对预测偏离度进行量化度量的一种精度,计算方式如式(15)所示[27]。

$$Score_{\mathrm{RP}}=\sum_{i=1}^{k}dist(\boldsymbol{pl}_i,\boldsymbol{l}_i)/K \tag{15}$$

由于每个位置是大小相等的网格,因此可以将每个网格视为坐标轴上1×1的单元格。故式(15)中的距离函数 $dist$ 采用曼哈顿距离计算两个网格之间的距离。在预测结果非绝对精确时,设定相对预测精度可以通过其与真实位置的距离偏差来选择相对精确的预测点,$Score_{\mathrm{AP}}$ 越低,表示预测精度越高。

4.3　轨迹数据预处理与样本集生成

从原始数据中,提取出在北京的13 101条轨迹,使用6位Geohash编码对轨迹进行网格化(约600 m×600 m网格),共生成14 777个不同的位置。将经纬度轨迹转化为位置序列并分段,设分段时间阈值为30 min,位置包含最小有效轨迹点数为10。

通过上述方式,将原始轨迹转化为21 740条位置序列。对提取出的位置序列,使用滑动窗口的方式生成样本集。设滑动窗口长度 $len=8$,滑动步长 $step=3$,除去长度小于 len 的位置序列后,得到69 537个样本,随机取其中的5 000个样本作为测试集,其余数据作为训练集。轨迹预处理的参数选择总结如表2所示。

Table 2　Parameters of trajectory preprocess

表2　轨迹预处理参数

参数名	参数值	参数说明
δ	10	最小有效轨迹点数
γ/min	30	分段时间阈值
len	8	滑动窗口长度
$step$	3	滑动步长

4.4　LDRM模型性能分析

通过预处理与样本集生成,将原始Geolife数据转化为轨迹位置序列训练集与测试集。在进行预测之前,需要使用轨迹位置序列训练集对LDRM位置降维模型进行训练。由于数据集中存在14 777个位置,即输入的位置one-hot向量的维度 $n=14\,777$。设LDRM降维得到的位置嵌入向量的维度为 m。位置嵌入向量维度 m 对预测结果有较大影响,过大的位置嵌入向量维度仍然会导致一定程度的维数灾难问题,过小的位置嵌入向量维度会造成大量的信息在降维中损失,都会导致预测维度下降。因此,将 m 的取值范围预设为50到500,步长为50。通过对训练集进行交叉验证计算绝对精度与相对精度,计算 m 的取值对LDRM-LSTM位置预测算法的效果的影响,实验结果如图6和图7所示。

由图6和图7可见,位置嵌入向量在100~150维之间时,预测结果与真实结果的绝对差距最小,相对偏离度最小,即绝对精度和相对精度均达到最佳,预

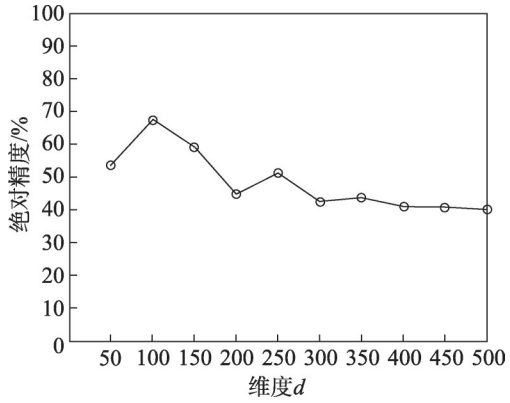

Fig.6 Absolute precision of LDRM-LSTM in different dimensions

图6 不同维度下LDRM-LSTM预测的绝对精度

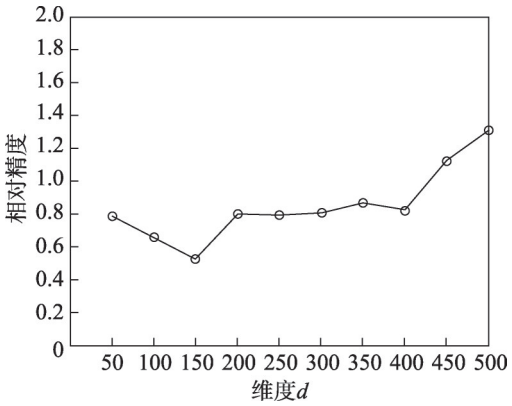

Fig.7 Relative precision of LDRM-LSTM in different dimensions

图7 不同维度下LDRM-LSTM预测的相对精度

测效果最好。由于绝对精度与相对精度相比更为重要,因此取绝对精度最高的100维作为降维后位置嵌入向量的维度。为了方便观察嵌入向量空间分布,将位置嵌入向量经过主成分分析(principal component analysis,PCA)降维到二维后,部分位置嵌入向量在二维向量空间中的分布如图8所示。每个位置点上的标签为各自的Geohash编码,图中越接近的两个空间向量,代表其在运动轨迹中越相关。

4.5 LSTM位置预测算法结果与分析

在确定了最优的嵌入向量维度并在将训练集与测试集中位置序列均转化为位置嵌入向量序列后,使用训练集中降维后的位置嵌入向量序列对LSTM神经网络进行训练,再使用训练后的模型对测试集

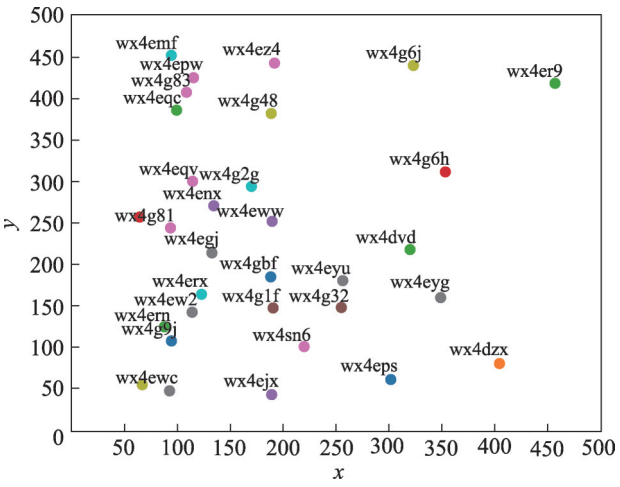

Fig.8 Graph of location embedding representation

图8 位置嵌入向量空间分布示意图

中的待预测的位置嵌入向量序列进行位置预测。将预测结果与真实值相对照,计算绝对精度与相对精度,并与HMM、MC、XGBoost(extreme gradient boosting)[28]、LSTM以及STS-LSTM等现有算法在同一样本集上进行对比。这些算法均为离线训练在线预测算法,离线训练时间和参数相关,在线预测时间复杂度几乎一致。实验结果如表3所示。

Table 3 Results of location prediction

表3 位置预测结果

Algorithm	Absolute precision/%	Relative precision
2阶MC	47.5	2.672 1
HMM	56.9	1.874 2
XGBoost	62.1	0.747 1
LSTM	37.8	2.351 6
STS-LSTM	60.2	0.691 3
LDRM-LSTM	**67.3**	**0.655 2**

由表3可见,LDRM-LSTM算法的绝对预测精度和相对预测精度远远高于未采用降维算法的LSTM模型,证明LDRM降维模型较好地解决了维度灾难问题。同时,LDRM-LSTM预测绝对精度和相对精度均高于所列出的经典位置预测算法,说明在解决长序列位置预测的问题上,LDRM-LSTM算法可以克服现有算法存在的弊端,得到较好的预测结果。

5 结束语

移动对象位置预测与运动模式挖掘一直是研究

的热点,可为各类基于位置的服务和应用提供重要支持,如智能交通系统、智能导航系统、路线规划等,具有很高的实用性和现实意义。传统的基于Markov概率模型的位置预测算法和基于神经网络的位置预测算法无法处理位置过多带来的维数灾难问题。提出了LDRM模型,通过神经网络挖掘移动对象轨迹数据中的隐含规律,将原始的高维one-hot向量转化为低维度的位置嵌入向量,再通过基于LSTM的移动对象位置预测算法进行位置预测。实验证明本文提出的LDRM-LSTM算法预测准确率高于经典的MC、HMM、RNN及XGBoost等算法。同时,算法采用离线训练在线预测的流程结构,使用历史数据训练的模型进行位置预测,在实际应用中有较高的价值。

References:

[1] Bao J, Zheng Y, Mokbel M F. Location-based and preference-aware recommendation using sparse geo-social networking data[C]//Proceedings of the 20th International Conference on Advances in Geographic Information Systems, Redondo Beach, Nov 7-9, 2012. New York: ACM, 2012: 199-208.

[2] Bao J, He T F, Ruan S J, et al. Planning bike lanes based on sharing-bikes' trajectories[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, Aug 13-17, 2017. New York: ACM, 2017: 1377-1386.

[3] Feng Z N, Zhu Y M. A survey on trajectory data mining: techniques and applications[J]. IEEE Access, 2017, 4: 2056-2067.

[4] Giannotti F, Nanni M, Pinelli F, et al. Trajectory pattern mining [C]//Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, Aug 12-15, 2007. New York: ACM, 2007: 330-339.

[5] Killijian M O. Next place prediction using mobility Markov chains[C]//Proceedings of the Workshop on Measurement, Privacy, and Mobility, Helsinki, Augu 13-17, 2012. New York: ACM, 2012: 3.

[6] Krumm J. A Markov model for driver turn prediction[C]// Proceedings of the Society of Automotive Engineers World Congress, Detroit, Apr 14-17, 2008. Warrendale: SAE World Congress, 2016: 1-7.

[7] Morzy M. Prediction of moving object location based on frequent trajectories[C]//LNCS 4263: Proceedings of the 21st International Symposium on Computer and Information Sciences, Istanbul, Nov 1-3, 2006. Berlin, Heidelberg: Springer, 2006: 583-592.

[8] 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, Feb 12-17, 2016. Menlo Park: AAAI, 2016: 194-200.

[9] Bengio Y, Simard P Y, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE Transactions on Neural Networks, 1994, 5(2): 157-166.

[10] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[11] Koren Y, Bell R M, Volinsky C. Matrix factorization techniques for recommender systems[J]. IEEE Computer, 2009, 42(8): 30-37.

[12] Xiong L, Chen X, Huang T K, et al. Temporal collaborative filtering with Bayesian probabilistic tensor factorization[C]// Proceedings of the SIAM International Conference on Data Mining, Columbus, Apr 29-May 1, 2010. Philadelphia: SIAM, 2010: 211-222.

[13] 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, Paris, Jun 28-Jul 1, 2009. New York: ACM, 2009: 637-646.

[14] 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 Symposium on Advances in Geographic Information Systems, Chicago, Nov 1-4. New York: ACM, 2011: 34-43.

[15] Morzy M. Mining frequent trajectories of moving objects for location prediction[C]//LNCS 4571: Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition, Leipzig, Jul 18-20, 2007. Berlin, Heidelberg: Springer, 2007: 667-680.

[16] Pei J, Han J W, Mortazaviasl B, et al. PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth[C]//Proceedings of the International Conference on Data Engineering, Heidelberg, Apr 2-6, 2001. Washington: IEEE Computer Society, 2001: 215.

[17] Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized Markov chains for next-basket recommendation [C]//Proceedings of the 19th International Conference on World Wide Web, Raleigh, Apr 26-30, 2010. New York: ACM, 2010: 811-820.

[18] Mathew W, Raposo R, Martins B. Predicting future locations with hidden Markov models[C]//Proceedings of the ACM Conference on Ubiquitous Computing, Pittsburgh, Sep 5-8, 2012. New York: ACM, 2012: 911-918.

[19] Al-Molegi A, Jabreel M, Ghaleb B. STF-RNN: space time features-based recurrent neural network for predicting people next location[C]//Proceedings of the IEEE Symposium Series on Computational Intelligence, Athens, Dec 6-9, 2016. Piscataway: IEEE, 2016: 1-7.

[20] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]//Proceedings of the Annual Conference on Neural Information Processing Systems, Montreal, Dec 8-13, 2014. Red Hook: Curran Associates, 2014: 3104-3112.

[21] Palangi H, Deng L, Shen Y L, et al. Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval[J]. IEEE/ACM Transactions on Audio, Speech and Language Processing, 2016, 24(4): 694-707.

[22] Visin F, Kastner K, Cho K, et al. ReNet: a recurrent neural network based alternative to convolutional networks[J]. arXiv: 1505.00393, 2015.

[23] Malhotra P, Ramakrishnan A, Anand G, et al. LSTM-based encoder-decoder for multi-sensor anomaly detection[J]. arXiv: 1607.00148, 2016.

[24] 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.

[25] Jin A, Cheng C Q, Song S H, et al. Regional query of area data based on geohash[J]. Geography and Geo-Information Science, 2013, 29(5): 31-35.

[26] Gers F A, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM[J]. Neural Computation, 2000, 12(10): 2451-2471.

[27] Qiao S J, Li T R, Han N, et al. Self-adaptive trajectory prediction model for moving objects in big data environment [J]. Journal of Software, 2015, 26(11): 2869-2883.

[28] Chen T Q, Guestrin C. XGBoost: a scalable tree boosting system[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, Aug 13-17, 2016. New York: ACM, 2016: 785-794.

附中文参考文献：

[25] 金安, 程承旗, 宋树华, 等. 基于Geohash的面数据区域查询[J]. 地理与地理信息科学, 2013, 29(5): 31-35.

[27] 乔少杰, 李天瑞, 韩楠, 等. 大数据环境下移动对象自适应轨迹预测模型[J]. 软件学报, 2015, 26(11): 2869-2883.

GAO Ya was born in 1994. She is an M.S. candidate at College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, and the member of CCF. Her research interests include location prediction of moving object and data mining, etc.

高雅(1994—),女,吉林松原人,南京航空航天大学计算机科学与技术学院硕士研究生,CCF学生会员,主要研究领域为移动对象位置预测,数据挖掘等。

JIANG Guohua was born in 1963. He is an associate professor and M.S. supervisor at Nanjing University of Aeronautics and Astronautics. His research interests include software testing and Internet of things engineering, etc.

江国华(1963—),男,江西婺源人,南京航空航天大学副教授、硕士生导师,主要研究领域为软件测试,物联网工程等。

QIN Xiaolin was born in 1953. He is a professor and Ph.D. supervisor at Nanjing University of Aeronautics and Astronautics, and the senior member of CCF. His research interests include spatial and spatio-temporal databases, data management and security in distributed environment, etc.

秦小麟(1953—),男,江苏南京人,南京航空航天大学教授、博士生导师,CCF高级会员,主要研究领域为空间与时空数据库,分布式数据管理与安全等。

WANG Zhongyu was born in 1994. He is an M.S. candidate at College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, and the member of CCF. His research interests include data driven fault detection in aircraft engine and data mining, etc.

王钟毓(1994—),男,江苏泰州人,南京航空航天大学计算机科学与技术学院硕士研究生,CCF学生会员,主要研究领域为航空发动机的异常检测,数据挖掘等。