计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.191000194

基于循环神经网络的轨迹压缩算法

励益韬　孙未未
复旦大学计算机科学技术学院　上海 201203
上海市数据科学重点实验室(复旦大学)　上海 201203
(yitaoli17@fudan.edu.cn)

摘　要　随着定位技术和存储技术的发展,海量的轨迹被人类记录。如何有效地压缩轨迹中最被人关注的空间路径信息并无损地将原始信息还原,引起了人们的广泛关注。轨迹压缩算法主要分为基于简化线段的压缩和基于路网的轨迹压缩两类,现有算法存在算法假设不合理、压缩能力差等缺点。文中根据路网中轨迹的分布特性以及循环神经网络对变长时序序列的建模能力,提出了基于循环神经网络的轨迹压缩算法,通过深度学习模型高效地概括轨迹分布,同时利用路网结构进一步缩小压缩空间,定量分析了不同输入对算法压缩比的影响。最后通过实验证明,基于循环神经网络的轨迹压缩算法不仅具有比现有算法更高的压缩比,还能支持未经过训练的轨迹数据的压缩;同时验证了终点信息如何对算法压缩比产生影响的假设。

关键词:轨迹压缩;循环神经网络;深度学习;轨迹建模
中图法分类号　TP301

Trajectory Compression Algorithm Based on Recurrent Neural Network

LI Yi-tao and SUN Wei-wei
School of Computer Science,Fudan University,Shanghai 201203,China
Shanghai Key Laboratory of Data Science,Fudan University,Shanghai 201203,China

Abstract　With the development of positioning technology and storage technology,massive trajectories have been recorded by humans. How to effectively compress the most interesting spatial path information in the trajectory and how to restore the original information has caused extensive research. The compression algorithm for trajectories is mainly divided into line-simplified compression and road-based trajectory compression. Existing algorithms have shortcomings such as unreasonable algorithm assumptions and poor compression capability. According to the distribution characteristics of trajectories in the road network and the probabilistic modeling ability of recurrent neural networks for variable-length time series,a trajectory compression algorithm based on recurrent neural network is proposed. The trajectory distribution is efficiently summarized by our algorithm,in which the compression space is further reduced by the road network structure. Meanwhile,the influence of different input on the compression ratio of the algorithm is quantitatively analyzed. Finally,the experiment proves that the trajectory compression algorithm based on recurrent neural network not only has a higher compression ratio than existing algorithms,but also supports the compression of untrained trajectory data,and demonstrates the compression ratio of the algorithm can be improved by using the time information.

Keywords　Trajectory compression,Recurrent neural network,Deep learning,Modeling trajectory

1　引言

在大数据时代,随着定位技术的发展和定位设备的普及,越来越多的先进技术可用于收集移动对象的定位,因此产生了海量的轨迹数据。以城市道路上的轨迹数据为例,其来源包含出租车、公交车、自行车、行人等种类繁多的移动对象。若不对原始轨迹数据进行压缩处理而直接使用,将对数据存储和通信成本带来极大的挑战,亟需轨迹压缩的技术手段来缩小数据的规模。同时,随着城市路网数据的日益完善,路网中的运动物体又严格受到道路拓扑结构的限制,利用路网的边序列来表征原始轨迹数据能有效缩减数据的大小,且能减少轨迹数据中定位设备精度带来的误差。

轨迹数据由时间信息和空间路径两个维度的数据组成[1]。轨迹压缩,即使用更少存储空间的数据信息来代表原始的轨迹时空信息,同时能利用被压缩后的数据信息无损或者在不影响数据使用的情况下,一定限度地有损还原出原始

到稿日期:2019-10-29　返修日期:2019-12-18　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(61772138);国家重点研发计划(2019YFB1704400)
This work was supported by the National Natural Science Foundation of China (61772138) and National Key Research and Development Program of China (2019YFB1704400).
通信作者:孙未未(wwsun@fudan.edu.cn)

的轨迹信息,以满足人们存储、查询、使用数据的需求。虽然轨迹中时间和空间这两个维度的数据存在一定的联系,但是为了研究简便,人们往往会将其分开,分别进行压缩。目前针对轨迹数据的压缩算法可以分为线段简化压缩方法和基于路网结构的压缩方法。绝大多数算法可以实现轨迹空间上的无损压缩,但在时间维度上都只能实现有损压缩。线段简化压缩方法基于欧氏空间,由于压缩时没有考虑结合限制轨迹拓扑结构的路网信息,因此使用时具有很大的局限性[2]。基于路网结构的压缩算法将原始的 GPS 定位轨迹转化为路网序列后进行进一步的压缩,存在以下几个缺陷。

(1)压缩基于一种人为的假设,如"轨迹总是沿道路的最短路径行驶"。然而,现实生活中道路拥堵、交通信号灯状况等情况,导致移动对象并不永远沿着最短路径前进。这种假设会影响数据的压缩效率。

(2)尝试统计轨迹的分布规律来缩短编码长度,但只考虑轨迹序列的二阶或三阶的依赖关系,缺少对轨迹全局分布规律的建模。

由于在实际应用中人们更关心轨迹的空间路径信息,且往往需要无损压缩来保证数据质量,因此本文只关注轨迹数据空间维度上的无损压缩。近年来,深度学习模型迅速发展,它能很好地表征数据分布特性,其中带长短期记忆门(Long Short-Term Memory, LSTM)的循环神经网络(Recurrent Neural Network,RNN)能有效地对变长序列的概率分布进行建模,且能很好地捕捉序列的"长期依赖关系"。本文提出的基于循环神经网络的轨迹压缩算法(Trajectory Compression Algorithm Based on Recurrent Neural Network,TCBR),利用改进的带 LSTM 门的循环神经网络来对轨迹数据的分布规律进行建模,能根据输入轨迹段预测下一条轨迹段;只需借助训练后的模型、轨迹的初始段信息(如每条轨迹的起点)以及模型预测错误的边序列的正确值即可还原数据,从而达到压缩轨迹数据的目的。其中,数据的压缩比与模型的预测准确率相关。本文主要有以下几点贡献:

(1)首次利用深度学习模型对路网中的轨迹进行压缩,且在压缩时不需要借助任何先验假设;

(2)定量地分析不同的输入对压缩算法的影响,同时通过优化存储方式提高了算法的压缩比;

(3)结合真实的轨迹数据进行大量实验,证明了该算法模型无论在训练过的数据上还是在未训练过的数据上,都能实现比现有方法更高的压缩比,体现了它的可行性和高效性。

本文第 1 节介绍了轨迹数据压缩的背景和现状;第 2 节分析了路网中的轨迹数据压缩问题的相关研究;第 3 节对本文研究问题的基本概念进行了定义;第 4 节对压缩算法展开详细介绍,包括算法中最为核心的深度神经网络的结构以及算法的压缩、解压缩流程;第 5 节分析不同输入对压缩算法的影响;第 6 节通过实验证明了 TCBR 方法的高效性;最后总结全文并对后续工作进行展望。

2 相关工作

近年来,轨迹压缩方法取得了长足的发展。根据是否借助路网信息,其分为两类。(1)不借助路网信息,直接对 GPS 点进行压缩,也称为线段简化压缩,目标是在规定的误差内减少 GPS 点的数目,典型算法有 Douglas-Peucker(DP)算法[4]。此类算法的主要思路为利用满足阈值的直线代替原来 GPS 点的折线,这样在原始的 GPS 点序列中,只需要存储彼此连线的夹角大于某个阈值的一部分点,从而减小轨迹数据的存储空间。然而,此类压缩算法存在以下两个缺点:1)压缩数据为有损压缩,可能会丢失或改变一些重要的轨迹信息;2)压缩比和轨迹的信息损失程度相关,压缩比越大,原始轨迹信息丢失得也越多,因此它的压缩比往往不能很高。同时,Han 等[2]已经证明了对原始的 GPS 点进行无损压缩很难减小数据量,将原始数据进行表示转换很有必要。(2)基于路网结构的压缩,即首先利用地图匹配算法将原始的 GPS 定位数据映射到对应的路网中,将轨迹的点序列转化为路网的边序列后再进行下一步的压缩。Cao 等[5]提出的 Non-material 算法首次利用路网结构对轨迹数据进行压缩,该算法将用路网表示的轨迹转化为路口的序列,从而达到压缩轨迹数据的目的。Lerin 等[6]利用最短路算法和链接算法对 Non-material 算法进行了改进。Kellaris 等[7]提出了地图匹配轨迹压缩算法(Map-Matched Trajectory Compression, MMTC),其主要思想是将轨迹压缩问题转化为函数最优化问题,找到一条最合适的轨迹,使得该轨迹尽可能用最短路径表示,并且近似轨迹与原始轨迹的相似度较高。Song 等[8]提出了基于路网的并行轨迹压缩(Paralleled Road-network-based Trajectory Compression,PRESS)算法,该算法分别压缩了轨迹的空间信息和时间信息,利用挖掘轨迹数据中的频繁序列模式来对轨迹进行编码,极大地减小了轨迹的压缩空间,同时降低了算法运行的时间复杂度。以上两种算法都是基于"轨迹沿着最短路径"的假设。Han 等[2]通过理论分析,证明了 Song[8]提出的将轨迹分别进行空间压缩和时间压缩的框架的合理性,同时提出了基于熵编码的 L&C 算法,论证了 L&C 压缩算法能达到假定条下的最优值。Kodie 等[9]在 Han[2]的研究的基础上,提出了 CiNCT 算法。

与此同时,大量研究者对轨迹建模进行了深入的研究,希望通过模型来概括轨迹数据的分布。Wu 等[12]利用多阶的马尔可夫链对轨迹的转移概率进行建模,但这无法捕捉轨迹的"长依赖"关系,同时也无法处理数据的稀疏问题。Zheng 等[10]提出利用贝叶斯反向强化学习方法对车辆轨迹进行建模,但其存在模型复杂、训练时间长等缺点;Ziebart 等[11]基于最大熵反向强化学习对轨迹进行建模,但模型的表达能力十分有限,不适用于复杂的大范围的路网条件;Wu 等[3]利用深度学习模型对轨迹进行建模,并通过实验论证了他们的方法在模型准确率、鲁棒性等指标上优于以上轨迹建模方法。

3 基本概念

本节将给出本文所需的相关基本定义。

定义 1(道路网络(road network,简称路网)) 将路网建模成一个有向图 $G(V,E)$,其中 V 是顶点(即道路路口)集合,E 是边(即路段)集合。对于 $\forall e\in E$,记 $e.s\in V$ 为路段的起点,$e.d\in V$ 为路段的终点。

定义 2(轨迹(trajectory)) 轨迹 T 受到路网的拓扑限制,可用连续的路网边序列 $T=e_1->e_2->\cdots->e_n$ 表示,且满足相邻边连续的性质,即 $\forall i, e_i.d=e_{i+1}.s$。构成该条轨

迹的路网边序列又称为该条轨迹的轨迹段，e_i 为一条轨迹段。

定义 3(轨迹概率模型) 轨迹 T 的分布概率为 $P(T)=P(e_1,e_2,\cdots,e_k)=P(e_1)\prod_{i=1}^{k-1}P(r_{i+1}|r_{1:i})$，每一条轨迹段与之前的轨迹段存在依赖关系。

定义 4(合法转移状态) 在路网 $G(V,E)$ 中，$\forall e_1,e_2\in E$，如果 $e_1.d=e_2.s$，则称 e_2 是 e_1 的合法转移状态，否则为非法转移状态。在一个城市路网中，任意一条边的合法转移状态一般不会超过 7 个。

4 算法的实现

本节将详细介绍 TCBR 算法中神经网络的结构以及 TCBR 压缩、解压缩数据的整体流程。

4.1 利用循环神经网络对轨迹进行建模

LSTM[13]门的 RNN 能很好地对序列进行建模，同时解决了 RNN 网络梯度灾难的问题，被广泛应用于自然语言处理领域。我们可以利用 LSTM 门的 RNN 对同样是时序序列的轨迹进行概率模型学习。然而，路网中的轨迹又完全不等同于自然语言领域的句子，因为每条边的合法转移状态是有限且极少的。在传统的 RNN 模型中，最后会通过分类的 softmax 层来输出每个状态是当前输入的下一个状态的概率。softmax 函数如式(1)所示，它是恒大于 0 的。

$$S_i=\frac{\exp(f_i)}{\sum_{j=1}^{C}\exp(f_j)} \tag{1}$$

限制状态空间的 RNN 模型[3](Constrained State Space RNN，CSSRNN)是一种基于路网拓扑结构特性而改进的深度学习模型，在保留 LSTM 特性的情况下，通过限制状态的 softmax 层来表征路网的拓扑结构，加快模型的运行速度，提升模型的学习能力。CSSRNN 的结构如图 1 所示。

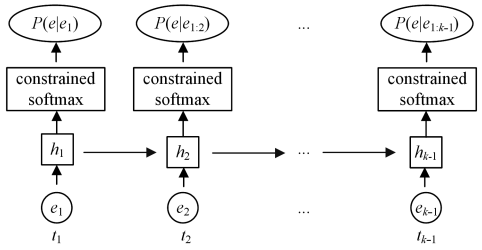

图 1 CSSRNN 的网络结构

Fig. 1 Structure of CSSRNN network

该模型的输入层为多维的嵌入(embedding)层，人为控制维度；输出层维度等于轨迹所在城市路网的边数，即状态空间的大小。T_i 时刻，轨迹中对应的轨迹段 e_i 的 embbeding 向量经过中间的带有 LSTM 门的隐层之后，通过限制状态的 softmax 层输出路网中每一条边是下一个轨迹段 e_{i+1} 的概率，其中概率值最大的网络节点所代表的路网边即为模型预测的下一条轨迹段。

CSSRNN 与传统 RNN 的最大不同就是将传统的 softmax 层改进为限制状态的 softmax 层。限制状态的 softmax 层利用路网的拓扑性质，在传统的 softmax 层前加相应的“掩码”(mask)：如果该状态是当前输入轨迹段的非法转移状态，则强制乘 0，否则乘 1。

形式化来说，t 时刻输入轨迹边 r_t，输出层中每一条路网边 i 对应的网络节点的概率输出如式(2)所示。

$$P(r'_{t+1}=i|r_{1:t})=\frac{\exp(W[i,:]h_t+b[i])\odot M_{r_t,i}}{\sum_{j=1}^{|E|}\exp(W[j,:])h_t+b[j])\odot M_{r_t,j}} \tag{2}$$

其中，h_t 表示前一层隐层的输出；$M_{r_t,i}$ 代表路网边 i 对于输入轨迹段 r_i 的“掩码”的值，如果路网边 i 与 r_i 相邻，$M_{r_t,i}$ 则为 1，反之为 0。

这样显式地将当前轨迹段的非法转移状态的概率置为 0 后，神经网络在训练过程中专注于当前输入轨迹段的合法转移状态，提高了网络学习轨迹数据分布的能力；同时又能加快模型的训练速度，因为非法转移状态对应的项被置为 0 后，它们的梯度不会向前传播，减少了网络在训练时的开销。

4.2 TCBR 算法的流程

图 2 展示了 TCBR 算法压缩和解压缩的流程。压缩过程主要分为 3 个阶段：地图匹配、模型训练和数据压缩。首先利用轨迹所在城市的路网信息和地图匹配算法，将轨迹由原始的 GPS 点坐标序列转化为路网边序列，之后利用轨迹数据对 CSSRNN 进行训练，直至模型收敛。

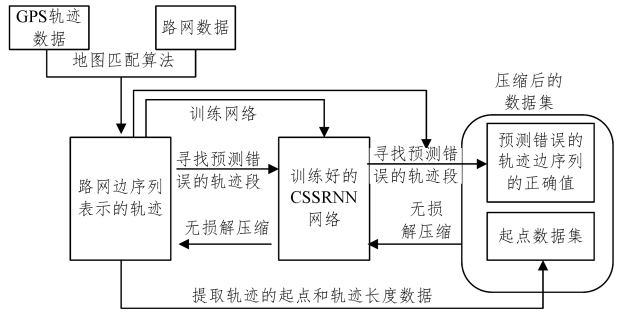

图 2 TCBR 算法模型的流程

Fig. 2 Framwork of TCBR algorithm

此时，CSSRNN 网络已经学习了轨迹数据集的分布。我们将原数据集输入该网络中，通过对比网络输出预测值与原数据找到网络预测错误的边序列，记录其正确值以及其在轨迹中的位置。具体来说，针对每一条轨迹 $T=(e_1,e_2,e_3,\cdots,e_k)$，依次向网络输入 $e_1,e_2,e_3,\cdots,e_{k-1}$ 这 $k-1$ 条路网边 ID，网络依次会输出 $e_2',e_3',\cdots,e_k'$。对比 e_i 和 e_i' 的值，如果两者不同，则将二元组 (e_i,i) 加入“预测错误的轨迹边序列的正确值”集合 S_{correct} 中，同时将每条轨迹的起点和轨迹长度信息加入集合 S_{start}，此时 S_{correct} 和 S_{start} 构成了被压缩的数据集。

解压缩时，针对每条轨迹 $T'=(e_1)$，一开始只有起点和轨迹长度信息。我们只需将其起点序列 ID 输入 CSSRNN 模型中，获得网络预测的下一条轨迹段的值。如果当前时刻的输出不在 S_{correct} 中，则把输出加入轨迹 T'，同时作为模型的输入来预测再下一个轨迹段；反之则用正确值替代输出值，将其加入轨迹 T' 中，并将其作为模型下一时刻的输入。如此循环，直到轨迹 T' 的长度等于之前记录的轨迹长度，将 T' 加入解压缩的轨迹集合中，转而对下一条轨迹采用同样的方式进行解压缩，最终无损地还原出原始轨迹数据。

5 加入终点信息的压缩方法

本节探讨在训练模型时加入终点信息的情况，同时介绍压缩后的数据的存储形式。

在训练 TCBR 算法模型时，可以将轨迹的其他信息作为

模型输入的一部分,如轨迹的终点信息,那么模型学习的轨迹概率分布就会变为式(3),即在既定轨迹终点"d"的情况下的概率分布。

$$P(T|d)=P(e_1|d)\prod_{i=1}^{k-1}P(e_{i+1}|e_{1,i},d) \tag{3}$$

从信息熵的角度来看,由于条件熵不大于非条件熵,这样的概率分布相比无终点信息的概率分布具有更小的不确定性,更容易被模型学习,模型可能具有更高的预测准确率。但是这也意味着每条轨迹的终点信息均需要被存储,可能导致被压缩的数据空间变大。后文将定量地探究加入轨迹的终点信息对压缩比的影响。

5.1 加入终点信息对压缩比的影响

表 1 列出了后文所需的相关符号的含义。

表 1 符号含义

Table 1 Frequently used notations

符号	含义
$\|E\|$	路网边的数目
T	数据集的轨迹数目
S	数据集的轨迹段数目
a	数据集的轨迹平均长度
N	数据集的最大轨迹长度
F	数据集中,终点在单条轨迹中出现的最大次数
P	不加入终点信息的模型预测错误率
P'	加入终点信息的模型预测错误率

对于不加入终点信息的轨迹压缩算法,它的起点数据集的比特数如式(4)所示,每条轨迹只需要存储起点轨迹段的原始 ID 和轨迹长度的二进制数。对于模型预测错误的轨迹段,记录它们的真实值,其占用存储空间的比特表示如式(5)所示。对于每条预测错误的轨迹段,我们需要记录其正确的 ID 和它在轨迹中的位置,但是不必直接存储正确轨迹段的 ID,只需要存储它在前一条轨迹段的邻接表中的 ID。在解压缩过程中,可以借助已知路网信息还原其轨迹段 ID。由于在实际路网中,一条边的合法转移状态不会超过 8 个,因此通常只需要 3 个比特的存储空间,远远小于$\lceil \log_2|E|\rceil$。这一转换也提高了模型的压缩性能。

$$S_1=T*(\lceil \log_2|E|\rceil+\lceil \log_2 N\rceil) \tag{4}$$

$$S_2=P*(S-T)*(3+\lceil \log_2 N\rceil) \tag{5}$$

对于加入终点信息的轨迹压缩算法,它的起点、终点数据集的比特数如式(6)所示,每条轨迹需要存储起点、终点轨迹段的 ID 编号以及终点在轨迹中出现的次数,其中终点在轨迹中出现次数的最大值为 F,F 远小于轨迹长度。利用 F 和终点 ID 能隐式地表征轨迹的长度,从而减小数据集的存储空间。对于预测错误的轨迹段,我们记录它们的真实值,其占用存储空间的比特表示如式(7)所示。

$$S_1'=T(2\lceil \log_2|E|\rceil+\lceil \log_2 F\rceil) \tag{6}$$

$$S_2'=P'(S-T)(3+\lceil \log_2 N\rceil) \tag{7}$$

如果加入终点信息的轨迹压缩算法的压缩性能优于不加入终点信息的压缩算法,则必须满足 $S_1'+S_2'<S_1+S_2$。根据 $S=aT$ 的性质,对其进行等式展开后发现,当 P 和 P' 满足不等式(8)时,加入终点信息的轨迹压缩模型的压缩性能更优,反之则不加入终点信息的轨迹模型的压缩性能更好。

$$P-P'>\frac{\lceil \log_2|E|\rceil+\lceil \log_2 F\rceil-\lceil \log_2 N\rceil}{(a-1)(3+\lceil \log_2 N\rceil)} \tag{8}$$

在定量比较是/否加入终点信息的压缩方法所占用的存储空间后,我们可以直接根据数据集参数以及训练出的模型的准确率情况,直接选择较优的模型进行压缩。

6 实验结果与分析

6.1 实验设置

本次实验使用葡萄牙波尔多(Porto)市和上海市的出租车轨迹数据集。波尔多市数据包含了 442 辆出租车自 2013 年 1 月 7 日到 2014 年 6 月 30 日约 1.8 GB 的行驶轨迹数据,上海市数据包含了 13 650 辆出租车自 2015 年 4 月 1 日至 2015 年 4 月 10 日约 16 GB 的数据。我们利用 OpenSrteetMap 上开源的地图信息和基于隐马尔可夫模型的地图匹配算法[14]将原始 GPS 点序列转化为对应的路网边序列;同时在每个城市中划分一块大区域和小区域,小区域内对应的轨迹数据分别标为 PT_{small} 和 SH_{small},大区域内对应的轨迹数据分别为 PT_{large} 和 SH_{large}。数据集的详细信息如表 2 所列。我们可以看到,小区域内的轨迹数目较少,平均长度不如大区域,但是拥有更加稠密的轨迹。

表 2 实验数据集的统计信息

Table 2 Statistics of datasets

	轨迹数	轨迹段数	轨迹平均长度	路网边数	每条路网边经过的轨迹数	路网节点数	单节点最大合法转移状态数
PT_{small}	486 268	18 327 682	37.69	6 117	79.5	3 182	4
PT_{large}	859 195	37 432 235	43.57	40 267	21.3	18 157	6
SH_{small}	757 032	19 147 708	25.29	8 075	93.8	3 632	5
SH_{large}	3 709 666	101 188 905	27.28	60 200	61.6	28 620	6

为了最大程度地学习整个轨迹数据集的数据分布规律,将训练集设置为全体数据集,选择 TensorFlow-gpu r1.6.0 作为深度学习模型的框架。实验的硬件配置如下:CPU 为 Intel Xeon E5-2643,内存大小为 256 GB,GPU 为 Nvidia GeForce TITAN X,显存大小为 12 GB,训练时间规定为 1.5 h。

实验过程中运行如下 8 种算法:1)TCBR_without,不输入终点信息的 TCBR 算法;2)TCBR_with,输入终点信息的 TCBR 算法;3)L&C 算法,由 Han[2] 提出,是由打标签(label)和编码(coding)两个步骤组成的熵编码算法,他证明了在 L&C 算法框架下该算法能达到理论最优压缩比;4)Zip,该算法是计算机压缩软件中最常用的算法;5)7Z,计算机压缩软件中压缩比最高的压缩算法;6)L&C+7Z,对 L&C 算法压缩后的文件用 7Z 压缩方法压缩;7)TCBR_without+7Z,用 7Z 压缩方法对不输入终点信息的 TCBR 算法压缩后的文件进行压缩;8)TCBR_with+7Z,用 7Z 压缩方法对输入终点信息的 TCBR 算法压缩后的文件进行压缩。

6.2 压缩性能分析

本次实验中的压缩比计算公式为:

$$压缩比=\frac{S_{原文件大小}}{S_{压缩文件大小}}$$

其中，$S_{原文件大小}$为转化为序列边后的轨迹数据集的二进制文件大小。$S_{压缩文件大小}$为随着数据集大小变化而变化的压缩后的文件的大小，在 TCBR 算法中其为 4.2 节中的 $S_{correct}$ 和 S_{start}，但不包括 TCBR 网络模型的大小。其原因在于：1）神经网络存储的内容实际就是训练后的网络参数，这只与路网的大小以及人为设计的超参数有关，在路网固定且神经网络超参数不变的情况下，无论轨迹数据变得多么大，模型的大小始终不会改变；2）由于路网中的轨迹受道路拓扑和环境的约束，同时同一物体能产生多条轨迹，同一路网下不同时期的数据在分布规律上具有相似性，已训练好的模型依然可以对同一路网下的其他轨迹进行压缩，具有普遍应用性。正因为以上两个特点，我们可以将其等同为实现压缩算法的编程代码，不必加入压缩比计算中。

实验得出不同算法的压缩比如图 3 所示。由图 3 可知，无论是否将终点加入训练中或使用什么类型的数据集，使用 TCBR 算法的压缩效果均好于其余任意单一的方法。其中，加入终点信息的算法模型的压缩比好于对比算法以及组合算法。同时，随着轨迹平均长度的增加，算法的压缩比也会增加；使用 7Z 压缩算法可以进一步提升 TCBR 方法的压缩比。

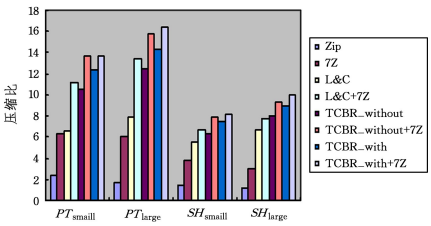

图 3 不同算法的压缩比

Fig. 3 Compression ratio of different algorithms

我们同时比较了 TCBR_without，TCBR_with 和 L&C 3 种算法的运行时间，结果如表 3 所列。

实验中，TCBR 的压缩时间始终长于 L&C 算法，这是深度学习框架的特性导致的。但是对比数据大小和对应的算法运行时，间可以发现，TCBR 算法和 L&C 算法的压缩时间和轨迹数目都呈线性关系。考虑到目前分布式技术和并行计算技术的日益发展，TCBR 算法在运行时间上的差异并不会降低它高压缩比带来的高效性。同时，带终点信息的 TCBR 模型由于需要将终点、起点信息同时作为网络输入，因此它会有更多的模型参数，这会增加网络前向计算开销，略微降低模型的压缩速度。

表 3 不同压缩算法的压缩时间

Table 3 Compression time of different algorithms

（单位：s）

算法	PT_{small}	PT_{large}	SH_{small}	SH_{large}
TCBR_without	238.39	456.84	324.48	2 036.22
TCBR_with	241.14	465.96	333.31	2 184.41
L&C	9.40	19.87	10.32	60.14

6.3 未训练数据的压缩比分析

本节通过实验探究训练好的 TCBR 模型是否能掌握未训练的数据的轨迹转移概率分布，分析其在未训练的数据下的压缩效果。

我们将 PT_{large} 的 859 195 条轨迹按照时间顺序排序，分别取总数据量的前 10%、前 20%、前 40%、前 60%、前 80%和 100%，形成 6 个数据集。利用前 10%的数据训练带终点信息的 TCBR 算法模型，再利用训练好的 TCBR 模型对以上 6 个数据集进行压缩，同时将 L&C 算法作为对比算法，实验结果如图 4 所示。

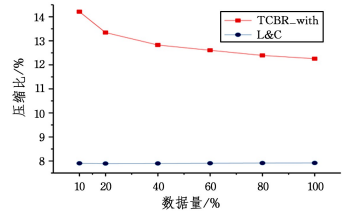

图 4 未训练数据下 TCBR 的压缩比

Fig. 4 Compression ratio of TCBR under incremental data

通过分析发现，随着未训练数据的增多，TCBR 算法的压缩比略有下滑，但是整体压缩比还是高于 12%，且压缩比下降趋势趋于平缓，远高于 L&C 算法的 7.9%。可以看到，TCBR 算法模型虽然没有学习后续 90%数据的分布规律，但是依然有着出色的压缩比。这也从侧面反映了同一路网下不同时间段的轨迹数据具有相似的分布规律。

同时，这论证了在同一路网下，TCBR 算法支持神经网络未训练数据的压缩。即使需要压缩的数据量增加到训练数据的 10 倍，其依然能保持良好的性能。TCBR 算法模型的参数并不需要实时根据输入轨迹数据进行训练更新，这可以很好地弥补深度学习模型训练收敛慢带来的影响。

6.4 终点信息对 TCBR 算法影响的分析

本次实验探究加入终点信息对 TCBR 算法的影响以及它的应用场景。

我们根据轨迹长度，从 PT_{largre} 的 859195 条轨迹中提取 4 个数据集，即 S_1，S_2，S_3，S_4，分别只包含长度为 1～20，21～40，41～60，61～80 的轨迹数据。对于每个数据集，我们单独训练输入终点信息和不输入终点信息的 TCBR 模型，并压缩原数据计算压缩比。8 种算法模型的预测错误率如表 4 所列，其中“理论阈值”等于不等式(8)右侧的值。实验结果如图 5 所示。

表 4 TCBR 模型的预测错误率

Table 4 Performance of TCBR models

（单位：%）

数据集	不带终点信息的错误率 P	带终点信息的错误率 P'	$P-P'$	理论阈值
S_1	9.80	1.85	7.95	10.77
S_2	8.23	2.33	5.9	4.73
S_3	6.57	2.32	4.25	3.00
S_4	6.03	2.48	3.55	1.77

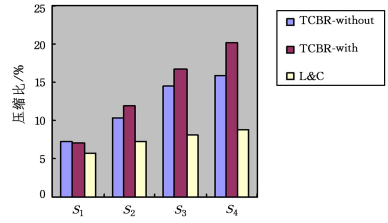

图 5 轨迹长度不同的数据集下的算法压缩比

Fig. 5 Compression ratio on different datasets

实验显示，当轨迹的线段平均长度较小时，TCBR-without 算法的压缩比优于其他两种算法。当轨迹平均长度增加时，TCBR-with 算法的性能更优。同时，当 $P-P'$ 的值小于理论阈值时，TCBR-without 的压缩比最高；大于理论阈值时，TCBR-with 模型的压缩比更优，证明了不等式(8)的正确性。

结合实验内容和不等式(8)，当数据集的轨迹平均长度很小时，我们推荐采用不带终点信息的 TCBR 算法；当数据集的轨迹平均长度变大时，将终点信息加入 TCBR 的模型训练中会得到更高的压缩比。

结束语 本文从轨迹建模的角度出发，研究了路网中轨迹空间路径压缩的问题，提出了基于 RNN 的压缩算法，通过深度神经网络对轨迹分布建立模型，解决了轨迹压缩的问题。同时，通过实验验证了本文算法的压缩比远优于其他的压缩算法，在压缩时间上呈线性复杂度，在同一路网下的未训练数据上压缩性能依旧出色，体现了其在实际应用中的可适用性。最后，指出了不带终点信息的和带终点信息的 TCBR 算法模型的应用场景。

在接下来的工作中，我们将研究如何将路网中轨迹的时间信息加入神经网络训练中，设计一套在时间和空间维度上都达到无损的轨迹压缩算法。

参 考 文 献

[1] ZHENG Y. Trajectory Data Mining: An Overview[J]. ACM Transactions on Intelligent Systems and Technology, 2015, 6(3):1-41.

[2] HAN H Y, SUN W W, ZHENG B H. COMPRESS: A Comprehensive Framework of Trajectory Compression in Road Networks[J]. ACM Transactions on Database Systems, 2017, 42(2):1-49.

[3] WU H, CHEN Z Y, SUN W W, et al. Modeling trajectories with recurrent neural networks[C]// Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17). AAAI Press.

[4] DOUGLAS D H, PEUCKER T K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature[J]. Cartographica: The International Journal for Geographic Information and Geovisualization, 1973, 10(2):112-122.

[5] CAO H, WOLFSON O. Nonmaterialized motion information in transport networks[M]// Database Theory - ICDT 2005. Berlin: Springer, 2005.

[6] LERIN P M, YAMAMOTO D, TAKAHASHI N. Encoding travel traces by using road networks and routing algorithms [M]// Intelligent Interactive Multimedia: Systems and Services. Berlin: Springer, 2012.

[7] KELLARIS G, PELEKIS N, THEODORIDIS Y. Map-matched trajectory compression[J]. Journal of Systems & Software, 2013, 86(6):1566-1579.

[8] SONG R, SUN W, ZHENG B, et al. PRESS: a novel framework of trajectory compression in road networks[J]. Proceedings of the Vldb Endowment, 2014, 7(9):661-672.

[9] KOIDE S, TADOKORO Y, XIAO C, et al. CiNCT: compression and retrieval for massive vehicular trajectories via relative movement labeling[C]// IEEE International Conference on Data Engineering (ICDE2018). IEEE, 2017.

[10] ZHENG J C, NI L M. Modeling heterogeneous routing decisions in trajectories for driving experience learning[C]// UbiComp. New York, USA: ACM Press, 2014:951-961.

[11] ZIEBART B D, MAAS A L, DEY A K, et al. Navigate like a cabbie: probabilistic reasoning from observed context-aware behavior[C]// International Conference on Ubiquitous Computing. ACM, 2008.

[12] WU H, MAO J Y, SUN W W, et al. Probabilistic robust route recovery with spatio-temporal dynamics[C]// Acm Sigkdd International Conference on Knowledge Discovery & Data Mining. ACM, 2016.

[13] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8):1735-1780.

[14] RAYMOND R, MORIMURA T, OSOGAMI T, et al. Map matching with hidden markov model on sampled road network[C]// International Conference on Pattern Recognition. 2012.

[15] ZHENG Y. Map-matching for low-sampling-rate gps trajectories [C]// Acm Sigspatial International Symposium on Advances in Geographic Information Systems. DBLP, 2009.

[16] ZHANG H Y, WU H, SUN W W, et al. Deeptravel: a neural network based travel time estimation model with auxiliary supervision[J]. arXiv: 1802.02147, 2018.

[17] WU J G, QIAN K Y, LIU M, et al. Hybrid trajectory compression algorithm based on multiple spatiotemporal characteristics [J]. Journal of Computer Applications, 2015, 35(5):1209-1212.

[18] SHAN Z Q, WU H, SUN W W, et al. COBWEB: a robust map update system using GPS trajectories[C]// Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 2015.

[19] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality [J]. Advances in Neural Information Processing Systems, 2013, 26:3111-3119.

[20] LUO W M, TAN H Y, CHEN L, et al. Finding time period-based most frequent path in big trajectory data[C]// Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. ACM, 2013.

LI Yi-tao, born in 1994, postgraduate. His main research interests include spatiotemporal data mining and so on.

SUN Wei-wei, born in 1973, Ph.D, professor, is a senior member of China Computer Federation. His main research interests include big spatiotemporal data and so on.