

基于 Transformer 的路网轨迹重建方法

梅宇生^{1,2}, 赵卓峰^{1,2}

(1. 北方工业大学信息学院, 北京 100144; 2. 大规模流数据集成与分析技术北京市重点实验室(北方工业大学), 北京 100144)

摘要: 轨迹重建是针对低采样轨迹数据进行轨迹补充还原的一类轨迹数据处理研究。为了提高轨迹重建的准确性, 一些工作通过引入 Seq2Seq 等深度学习模型来提升轨迹重建的效率与精度, 但由于现有工作忽略了轨迹间的长距离依赖问题, 导致轨迹还原中还存在准确率不高等问题。本文提出一种基于 Transformer 的轨迹重建模型 ZTrajRec (Zero-based trajectory recovery), 通过 Transformer 编码器捕获轨迹间的长距离依赖, 注意力机制用于当前轨迹和历史轨迹相似性查询来进行轨迹在路网上的重建。实验结果表明, 在真实北京出租车数据集上, ZTrajRec 比基准模型最好效果在召回率上提升 3%~4%。本文最后对重建结果进行了可视化分析以展示其合理性。

关键词: 轨迹重建; 路网; 序列到序列模型; Transformer

中图分类号: TP18 **文献标志码:** A

Map-Constrained Trajectory Recovery Mechanism Based on Transformer

MEI Yusheng^{1,2}, ZHAO Zhuofeng^{1,2}

(1. School of Information, North China University of Technology, Beijing 100144, China; 2. Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data (North China University of Technology), Beijing 100144, China)

Abstract: Trajectory reconstruction is a research field for trajectory restoration of low-sampling rate trajectory data. In recent years, in order to improve the accuracy of trajectory reconstruction, some work used deep learning models such as Seq2Seq to improve the efficiency and accuracy of trajectory recovery. However, most of the existing work ignores the long-distance dependencies between trajectory points, resulting in poor accuracy for trajectory reconstruction. Therefore, this paper proposes a trajectory recovery model, called ZTrajRec (Zero-based trajectory recovery) based on Transformer, which captures the long-distance dependency between trajectories through Transformer encoder, and uses the attention mechanism to take into account the similarity between current trajectory and historical trajectories to reconstruct the trajectory directly on the road network. Experimental results show that, on the real Beijing taxi dataset, ZTrajRec improves the recall rate by 3%–4%, compared to the results of the benchmark models. Finally, the result is visually analyzed to demonstrate its plausibility.

Key words: trajectory recovery; road network; Seq2Seq model; Transformer

引言

随着空间位置采样技术和移动计算设备的发展,大量空间轨迹数据得以产生,同时也伴随着相关轨迹数据应用的提出^[1]。例如,旅行时间估计^[2-4]、轨迹表征学习^[5-6]、交通预测^[7-9]以及驾驶员行为分析^[10]等。在之前相关工作^[11-12]中提到,为了使这些应用获得更好的效果,通常需要高采集率的轨迹数据。然而在实际中,部分设备产生的轨迹一般为低采集率。例如,出租车GPS设备,为节省能源的消耗,通常每2~6 min记录一次轨迹数据^[13]。针对上述轨迹低采集率问题,目前有两种主流方法:(1)两阶段转换法;(2)一阶段转换法。

两阶段转换法的主要做法是,首先对数据从低采集率轨迹转换为高采集率轨迹,再对生成后的高采集率轨迹进行路网匹配,以获得最后的路网匹配轨迹。这类方法中可以通过线性插补^[14]进行补全。在插补过程中,假定车辆一直保持匀速,并使行驶方向保持不变,通过在其相邻轨迹点进行补全即可获得高采集率轨迹数据。但这种方法因并没有考虑轨迹的复杂性以及路网信息,导致补全信息精度较差。针对线性插补的不足,Wang等^[15]提出深度混合轨迹恢复(Deep hybrid trajectory recovery model, DHTR)模型,通过将子序列到序列(Subsequence-to-sequence, Subseq2seq)与Kalman滤波集成用于第一阶段生成高采集率轨迹。首先通过模型进行第一阶段,对低采集率轨迹数据生成高采集率轨迹数据;再进行第二阶段,通过基于隐马尔科夫模型(Hidden Markov model, HMM)^[16]算法将高采集率轨迹还原到路网上,以获得高采集率的路网匹配轨迹。

而一阶段转换法则是通过给定低采集率轨迹直接对其进行处理,生成高采集率的路网匹配轨迹,而无需对原始轨迹进行多阶段处理。这类方法包括近期Ren等^[11]提出的地图约束的轨迹恢复(Map-constrained trajectory recovery, MTrajRec)模型。MTrajRec通过门控循环单元(Gated recurrent unit, GRU)对轨迹进行编码解码。在模型解码过程中,采用多任务学习方法,即同时输出两类结果:(1)当前轨迹点在路网上的路段;(2)此路段上行驶距离比例,从而获得路网上的轨迹点。以此确保模型能够一阶段直接生成路网匹配的高采集率轨迹。但是,针对低采集率数据,相关现有工作在轨迹重建问题上仍有一些改进的地方。传统模型^[15]在轨迹重建问题上,采取先对轨迹进行还原,再通过路网匹配算法将还原好的轨迹点匹配到路网上。这类方法不足的是,在第一阶段生成高采集率数据时,并未考虑路网信息,导致其还原的轨迹在准确率上效果较差。而MTrajRec模型直接将轨迹还原到路网上,虽然避免了后续路网匹配过程,但其采用GRU模型并没有解决轨迹在复杂情况下的轨迹长距离依赖问题,尤其是轨迹在复杂路网上多次变换方向。

为解决上述问题,本文提出了基于Transformer及Rezero^[17]架构的ZTrajRec模型,其包含编码器、注意力和解码器3个模块。编码器模型将输入的轨迹信息编码为上下文向量,并作为注意力模块的输入,用于轨迹重建任务;注意力模块计算当前轨迹与历史轨迹之间的相似度,然后选择是否将其用作辅助输入;解码器模块将低采集率轨迹还原到目标采样率,并同时进行了轨迹的路网匹配,保证输出轨迹约束在路网上。对于轨迹不同位置的处理,本文采用区域网络化处理来进行表示。

本文的主要贡献包括:(1)提出ZTrajRec轨迹重建模型,将Transformer模型应用于轨迹重建任务;融合提出的多头注意力模块用作辅助输入;利用长短期记忆网络(Long short-term memory network, LSTM)自回归方式捕获高采集率轨迹间时空信息的依赖关系,逐步还原轨迹到路网上。(2)在真实北京出租车数据集上设计了实验评价方法,并进行了相关实验。实验结果表明本文ZTrajRec模型在评价指标上具有较高的准确性。

1 轨迹重建问题定义

本节将给出轨迹重建中涉及的概念和本文关注的轨迹重建问题的定义。

定义 1(路网): 路网为用 $G=(V, E)$ 表示的路网图, 其中 $E=\{e_1, e_2, \dots, e_m\}$ 为路口交点之间的路段, 而 $V=\{v_1, v_2, \dots, v_n\}$ 代表路段之间的交点。每一个 $e_i \in E$, 包含起始节点 $v_q \in V$, 以及终止节点 $v_p \in V$ 。

定义 2(轨迹): 轨迹数据定义为 $Q=\{q_1, q_2, \dots, q_n\}$, 其中轨迹点 $q_i=(\text{lat}_i, \text{lng}_i, t_i)$, 分别代表纬度、经度以及经过当前位置的时间。

定义 3(路网匹配的轨迹): 指经过 GPS 设备获取的轨迹数据后, 经过路网匹配算法(如 HMM^[16])后得到轨迹。若非必要说明, 轨迹数据代表直接从 GPS 设备获取的轨迹数据。而路网匹配的轨迹则为经过路网匹配算法后得到的轨迹数据, 其表示为 $\hat{S}=\{s_1, s_2, \dots, s_n\}$, 其中 $s_i=(e_i, r_i, t_i)$ 。 e_i 代表路网 G 中的路段, $r_i \in [0, 1]$ 代表当前路段上, 从起始节点开始到当前位置与路段长度的距离比例。

定义 4(采集率): 采集率指在轨迹中, 每个轨迹点之间的时间间隔倒数, 例如采集率为每 15 s。若从 GPS 设备中获取的轨迹数据时间间隔不一致, 则采用平均时间间隔。低采集率轨迹数据则代表在轨迹中轨迹点之间的时间间隔较长的轨迹数据。

定义 5(采集率为 ϵ 路网匹配的轨迹): 给定 ϵ , 则其轨迹表示为 $U=\{(e_1, r_1, t_1), (e_2, r_2, t_1 + \epsilon), \dots, (e_n, r_n, t_1 + (n-1)\epsilon)\}$ 。

定义 6(轨迹重建): 给定一个低采集率 ϵ_p 的 GPS 轨迹, 模型能够重建其采集率为 ϵ_q (高采集率) 的轨迹, 并且重建的是路网上的轨迹, 而无需后续再用路网匹配算法修订。这里给定的低采集率的 GPS 轨迹, 各轨迹点时间间隔可不一致。即给定轨迹数据 $Q=\{q_1, q_2, \dots, q_n\}$ 。经过模型输出有: $S=\{s_1, s_2, \dots, s_m\}$, 且满足 $n \ll m$ 。

2 ZTrajRec 模型

针对轨迹重建任务, 本文模型首先利用 Transformer 编码器层, 通过对历史轨迹点进行离散化处理并融合位置信息, 并对其编码器层进行调整采用 Rezero 架构, 使其捕获历史轨迹的时空信息以及轨迹间长距离依赖关系的同时, 避免加入过多轨迹中包含的无关噪声信息。

轨迹点的还原跟其相邻轨迹有着时空依赖关系, 但其相邻的轨迹点对当前轨迹点却有着不同程度的影响。因此, 在其编码器与解码器之间提出采用多头注意力机制, 使模型关注轨迹序列的不同部分并学习它们之间的依赖关系。通过有效地捕获历史轨迹对当前还原轨迹点的长期依赖关系, 使模型可以捕获复杂的模式, 以提高后续还原的准确率。

最后, 在解码器中为确保还原的轨迹约束在路网上, 利用多任务学习方法, 生成轨迹路段及其距离比例。首先轨迹路段间的还原不仅有着路网的约束, 同样也具有时空依赖关系, 如当前还原的轨迹路段只能是与自身或上一还原的轨迹路段, 而非与其具有连通性但非近邻的路段。其次还原的距离比例同样有着时空依赖关系, 其本身也可看作在当前路段上轨迹点的还原。因此, 对于轨迹路段及距离比例的还原, 本文提出采用 LSTM 模型对其进行时空信息捕获, 利用自回归方式对轨迹进行逐点重建。

2.1 模型架构

本文基于 Transformer 模型^[18], 提出了一种基于 Transformer 的 Seq2Seq 模型^[19]——ZTrajRec 来解决路网匹配的轨迹重建问题。ZTrajRec 模型架构如图 1 所示。本模型共包含 3 个模块: 编码器模块、多头注意力模块和解码器模块。

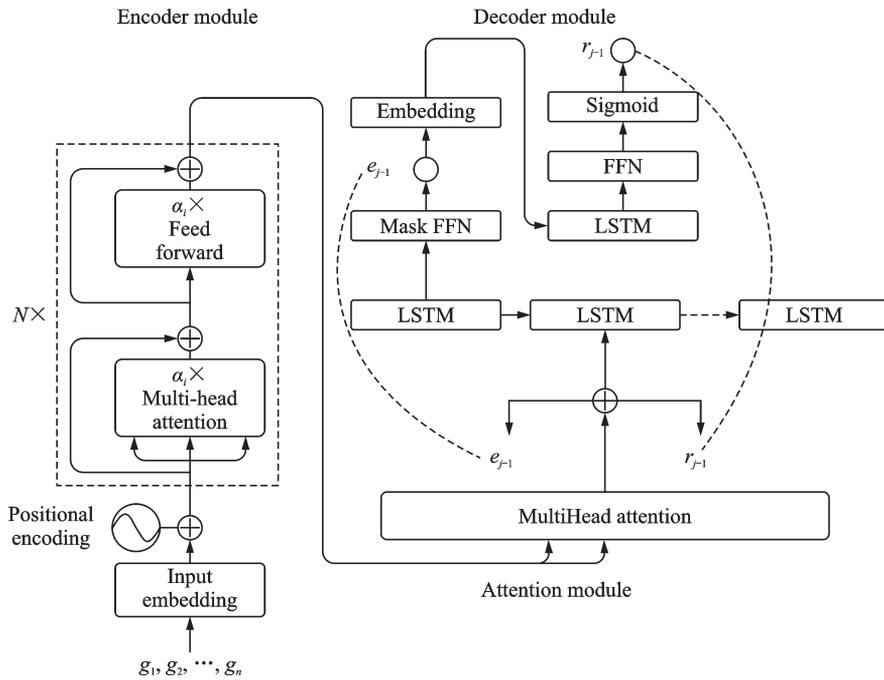


图1 ZTrajRec模型架构

Fig.1 Structure of ZTrajRec model

2.2 编码器模块

在给定低采集率GPS轨迹 $Q = \{q_1, q_2, \dots, q_n\}$ 进入编码器之前,需将经纬度数据离散化处理^[11]。首先将路网网格化处理,时间离散化处理,则轨迹点 $q_i = (\text{lat}_i, \text{lng}_i, t_i)$ 即可转化为 $g_i = (x_i, y_i, \text{tid}_i)$ 。其中, x_i, y_i, tid_i 分别代表轨迹纬度、经度所在网格的ID以及时间ID, $\text{tid}_i = \left\lfloor \frac{t_i - t_0}{\epsilon} \right\rfloor$, ϵ 为重建轨迹的高采集率。对 g_i 进行向量化处理有 $g'_i \in \mathbf{R}^{L \times d}$, 表达式为

$$g'_i = \text{Concat}(\text{Embedding}(x_i), \text{Embedding}(y_i), \text{tid}_i) \tag{1}$$

对于轨迹而言,每一个轨迹点在轨迹中的位置非常重要,代表着轨迹点序列信息。当长度为 L 的轨迹 G 进入编码器模块,其每一个轨迹点是利用注意力机制进行同时处理,而模型本身并没有保留轨迹位置信息的能力。因此,本文采用位置编码^[18],通过采用不同频率的 \sin, \cos 函数以确定轨迹点间的相对位置,并把得到的信息加到模型输入中,即

$$\begin{aligned} \text{PE}_{(\text{pos}, 2i)} &= \sin(\text{pos}/10\,000^{2i/d}) \\ \text{PE}_{(\text{pos}, 2i+1)} &= \cos(\text{pos}/10\,000^{2i/d}) \end{aligned} \tag{2}$$

式中: $\text{pos} \in [0, L/2)$ 为当前轨迹点的位置; d 则为 g'_i 的维度; $i \in [0, d/2)$, 其轨迹点的偶数位置和奇数位置分别对应上述的 \sin 和 \cos 函数。

则模型的输入即为

$$\hat{g} = g'_i + \text{PE}(g'_i) \tag{3}$$

编码器层包含注意力层^[18]以及前馈全连接层。注意力层给定长度为 L 的输入轨迹 $G \in \mathbf{R}^{L \times d}$, 有

$$\begin{cases} \text{Multi}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) \mathbf{W}^O \\ \text{head}_i = \text{Attention}(\mathbf{Q} \mathbf{W}_i^Q, \mathbf{K} \mathbf{W}_i^K, \mathbf{V} \mathbf{W}_i^V) \\ \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q} \mathbf{K}^T}{\sqrt{d}}\right) \end{cases} \quad (4)$$

式中： $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ 分别为轨迹输入 \hat{g} 的Query、Key、Value； $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V, \mathbf{W}^O$ 为模型要学习的参数； d 为 \hat{g} 的维度； h 为多头的个数。

前馈全连接层：每一层包含2个前向神经网络，并采用ReLU作为激活函数，表达式为

$$\text{FFN}(x) = \max(0, x \mathbf{W}_1 + \mathbf{b}_1) \mathbf{W}_2 + \mathbf{b}_2 \quad (5)$$

式中： $\mathbf{W}_1, \mathbf{W}_2$ 为当前层的要学习的参数； $\mathbf{b}_1, \mathbf{b}_2$ 为前向网络的偏置参数。

另外通过全连接层嵌入其他环境信息(如当日小时,是否为节假日),获得编码器输出 $s \in \mathbf{R}^{L \times d}$ 。为了模型的快速收敛,这里采用Rezero^[17]架构。通过对编码器中注意力层以及前馈全连接层加入学习参数 α ,且 α 初始化为0,如图1(a)编码器所示。这个方法的基本思想是编码器层中的注意力层、前馈全连接层在训练初期嘈杂,并会给后面计算带来不必要的复杂性,故在训练过程中,模型从简单开始,然后根据需要逐渐引入复杂性。若sublayer为当前注意力层与前馈全连接层组成的编码器层(见图1中(a)编码器),则编码器层的输出为

$$x_{i+1} = x_i + \alpha_i \text{sublayer}(x_i) \quad (6)$$

2.3 多头注意力模块

基于传统的Seq2Seq模型,在对轨迹重建时并未考虑历史轨迹对当前轨迹点重要程度。在室内轨迹自动恢复(Indoor trajectory automatic recovery, ITAR)模型^[20]还原轨迹中,作者提出了注意力机制去捕获全局信息,并通过消融实验验证了其有效性。本文基于多头注意力思想,在编码器输出进入解码器时先加入多头注意力机制。其机制主要作用是根据当前轨迹自动获取相关的历史轨迹信息,并利用多头机制,使模型从历史轨迹中选取不同维度,多头中的单头独立地获取输入的不同部分进行同时计算,最后在融合获得相关性信息,即

$$c_i = c_i^{(1)} \| c_i^{(2)} \| \dots \| c_i^{(H)} \quad (7)$$

式中：“ $\|$ ”为不同相关性信息拼接操作(Concatenation),每个相关性信息 $c_i^{(h)}$ ($h = 1, 2, \dots, H$)由编码器输出 s 与注意力得分得出,即

$$c_i^{(h)} = \sum_{j=1}^n \alpha_{i,j}^{(h)} s_j \quad (8)$$

其中注意力得分 $\alpha_{i,j}^{(h)}$ 由编码器输出 s_j 与当前解码器hidden计算得出,即

$$\alpha_{i,j}^{(h)} = \frac{\exp(u_{i,j}^{(h)})}{\sum_{j'=1}^N \exp(u_{i,j'}^{(h)})} \quad (9)$$

$$u_{i,j}^{(h)} = v^{(h)} \tanh(W_h^{(h)} h_i + W_s^{(h)} s_j) \quad (10)$$

式中： $W_h^{(h)}, W_s^{(h)}, v^{(h)}$ 均为模型要学习的参数； h_i 为解码器中的hidden； s_j 为编码器中的输出。

2.4 解码器模块

解码器通过自回归方式,逐点还原目标高采集率的路网轨迹 $U = \{(e_1, r_1, t_1), (e_2, r_2, t_1 + \epsilon), \dots, (e_n, r_n, t_1 + (n-1)\epsilon)\}$ 。解码器模块采用两个LSTM模型分别输出轨迹路段ID和路段中的行驶距离比例。解码器中路段hidden更新为

$$\mathbf{h}_i = \text{LSTM}(\mathbf{h}_{i-1}, \mathbf{e}_{i-1}, \mathbf{r}_{i-1}, \mathbf{c}_i) \quad (11)$$

式中: $i \in (1, 2, \dots, n)$ 为重建轨迹点的序号; \mathbf{e}_{i-1} 代表第*i*-1个还原轨迹路段的 embedding 向量; \mathbf{r}_{i-1} 代表*i*-1个轨迹路段的行驶距离比例; \mathbf{c}_i 为2.3节注意力模块输出; J 为路网G中全部路段的集合。最后再通过 softmax 层获取第*i*个轨迹路段 \hat{e}_i 。

$$p(\hat{e}_i | \mathbf{h}_i) = \frac{\exp(\mathbf{h}_i \cdot \mathbf{w}_j) \odot \mathbf{c}_i}{\sum_{j \in J} \exp(\mathbf{h}_i \cdot \mathbf{w}_{j'}) \odot \mathbf{c}_i} \quad (12)$$

式中: \mathbf{w}_j 为模型要学习的参数; \mathbf{c}_i 为文献[11]提出的约束条件。解码器中路段行驶比例 hidden 更新为

$$\mathbf{h}'_i = \text{LSTM}(\mathbf{h}'_{i-1}, \mathbf{e}_i, \mathbf{h}_i) \quad (13)$$

式中 \mathbf{e}_i 代表当前还原轨迹路段 embedding。则获取行驶距离比例 \hat{r}_i , 有

$$\hat{r}_i = \sigma(\mathbf{w}_j \mathbf{r}_i + b_j) \quad (14)$$

式中: σ 为 Sigmoid 激活函数; \mathbf{w}_j, b_j 为模型学习参数; \mathbf{r}_i 为解码器 hidden 中的第*i*个行驶距离比例。

约束条件:在 MtrajRec 模型中^[11],为了使模型能够更准确地还原轨迹路段,作者提出了一种基于欧式距离获取当前轨迹 GPS 点对于周边距离小于 50 m 路段 ID 的影响程度的方法,即 $f(d) = e^{-\frac{d^2}{\beta^2}}$ ($\beta = 15$) 对于在低采集率 GPS 轨迹 l 中,周边路段 ID 大于等于 50 m 的路段,其影响程度设置为 0,其余情况设置为 1。其中 $d_{j,l}$ 为当前 GPS 轨迹点 q_j 路网匹配到周边路段上点 q'_l 的欧式距离,用公式表示即为

$$\mathbf{c}_{j,l} = \begin{cases} f(d_{j,l}) & j \in g_i.\text{tid} \mid i \in [1, n], \text{且 } d_{j,l} < 50 \\ 0 & j \in g_i.\text{tid} \mid i \in [1, n], \text{且 } d_{j,l} \geq 50 \\ 1 & \text{其他} \end{cases} \quad (15)$$

损失函数:

(1)对于路段 ID,损失函数采用交叉熵损失函数,其中 \hat{e}_i 为预测路段 ID,即为

$$\mathcal{L}_{\text{eid}} = - \sum_{i=1}^N \sum_{j=1}^L \ln(P(\hat{e}_i | \mathbf{h}_i)) \quad (16)$$

式中: N 为重建轨迹的长度; L 为路段 ID 的个数。

(2)对于路段行驶比例,损失函数采用均方误差损失函数,即为

$$\mathcal{L}_{\text{rate}} = \sum_{i=1}^N \sum_{j=1}^L (r_i - \hat{r}_i)^2 \quad (17)$$

式中: r_i 为真实距离比例; \hat{r}_i 为预测距离比例。

结合两个损失函数,最终损失函数为

$$L = \mathcal{L}_{\text{eid}} + \lambda \mathcal{L}_{\text{rate}} \quad (18)$$

式中 λ 为可调参数,这里选取为 10。

3 实 验

3.1 数据集

本文采用 2012 年 10 月 1~14 日采集的北京出租车 GPS 轨迹数据,以及 OpenStreetMap 提供的路网信息。为获取高采集率轨迹数据,对原始数据进行线性插补^[14],以获取间隔为 15 s 的轨迹数据。为验证本模型的可扩展性,选取北京城市中心区域。选取的区域跨度为(39.909 5°, 116.429 2°)到(39.946 8°, 116.512 8°)的长方形区域,共包含 3 027 条路段。

实验轨迹数据通过 7:2:1 的比例划分为训练集、验证集以及测试集。为验证模型对轨迹重建能力,对获取低采集率的数据进行保持率为 {6.25%, 12.5%, 25%} 的比例采样。较小的保持率意味着模型要恢复更多的轨迹点。如保持率为 6.25% 的数据,模型要重建原有轨迹 93.75% 的数据。为验证模型是否真实重建路网匹配后的轨迹数据,本文采用 HMM^[16] 算法对原有 GPS 轨迹进行路网匹配,以获取标签数据。经过处理划分后的数据信息如表 1 所示。

表 1 数据集信息
Table 1 Dataset information

信息	路段数/ 个	训练集/ 条	验证集/ 条	测试集/ 条
14 天数据	3 027	165 995	47 045	23 466

3.2 评价指标

轨迹重建是把从低采集率的轨迹数据还原成路网匹配后的高采集率的轨迹数据。依照文献[11]所使用的指标,本文采用路段的准确率以及轨迹点的路网距离作为模型的评价指标。

RMSE 和 MAE: 均方根误差 RMSE 与平均绝对误差 MAE 是回归问题中经常使用测量真实值与预测值匹配程度的指标。对于模型重建的轨迹数据,采用 MTrajRec 模型^[11] 中轨迹点的距离误差进行计算。对于给定的真实轨迹 $\alpha = \{(e_1, r_1, t_1), (e_2, r_2, t_2), \dots, (e_m, r_m, t_m)\}$ 以及重建轨迹 $\hat{\alpha} = \{(\hat{e}_1, \hat{r}_1, \hat{t}_1), (\hat{e}_2, \hat{r}_2, \hat{t}_2), \dots, (\hat{e}_m, \hat{r}_m, \hat{t}_m)\}$ 。MAE 和 RMSE 评价指标计算公式分别为

$$\text{MAE} = \frac{1}{M} \sum_{j=1}^m |\text{dis}(a_j, \hat{a}_j)|$$

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{j=1}^m (\text{dis}(a_j, \hat{a}_j))^2} \quad (19)$$

$$\text{dis}(a_j, \hat{a}_j) = \min(\text{rn_dis}(a_j, \hat{a}_j), \text{rn_dis}(\hat{a}_j, a_j))$$

式中 $\text{rn_dis}(a_j, \hat{a}_j)$ 代表从路网上点 a_j 到点 \hat{a}_j 的行驶距离。因为路网结构为有向图,故 dis 计算两类双向数据,并取其中最小值作为行驶距离。

Recall 和 Precision: 为了进一步评价重建轨迹是否正确还原路段信息,采用召回率 Recall 和精确率 Precision 作为评价指标。给定真实轨迹经过的路段 $E = \{e_1, e_2, \dots, e_m\}$ 和重建轨迹 $\hat{E} = \{\hat{e}_1, \hat{e}_2, \dots, \hat{e}_m\}$, 其指标计算公式分别为

$$\text{Recall}(E, \hat{E}) = \frac{|E \cap \hat{E}|}{|\hat{E}|} \quad (20)$$

$$\text{Precision}(E, \hat{E}) = \frac{|E \cap \hat{E}|}{|\hat{E}|} \quad (21)$$

3.3 实验设置和基准模型

本文提出的 ZTrajRec 模型以及基准模型均使用 PyTorch 1.11.0 实现。网格选取大小为 50 m × 50 m。编码器与解码器的隐藏层的神经元个数均为 512。所有模型均由学习率为 0.001 的 Adam optimizer^[21] 经过 20 epochs 训练, batch size 选取为 128。实验设备为英特尔至强 Silver 4210 处理器和 24 GB Tesla P40 GPU。

本文采用 4 个模型作为轨迹重建的基准模型:

(1) Linear^[14] + HMM^[16]: 通过对低采集率的轨迹点进行线性插补获得高采集率的轨迹,再通过 HMM 算法匹配到路网上。

(2)DeepMove^[22]+Rule: 利用递归神经网络的历史注意力机制模型预测下一轨迹。通过利用DeepMove预测下一路段ID,并取每个路段的中间位置作为轨迹的重建。

(3)T2vec^[5]+Decoder: 一个基于轨迹相似性计算的Seq2seq模型。通过加入解码器使模型能够进行输出轨迹路段,以及路段中的具体位置。

(4)MTrajRec^[11]: 一个利用轨迹数据,以端对端,无需路网匹配轨迹还原的Seq2seq模型。其通过多任务学习解决GPS轨迹重建并直接还原到路网,并通过注意力机制以及路网相关信息进一步提升性能。

3.4 消融实验结果及分析

为验证推出的多头注意力机制对模型的效果验证,通过对注意力模块进行拆分,从多个评价指标体现注意力机制的贡献。

ZTrajRec-noAttn为消除编码器与解码器中间的多头注意力模块。采用解码器直接对编码器结果进行轨迹重建,则解码器hidden表达式为

$$h_i = \text{LSTM}(h_{i-1}, e_{i-1}, r_{i-1}) \tag{22}$$

实验通过Recall、Precision、MAE和RMSE四个指标,针对不同的数据保持率,对本文提出的模型ZTrajRec与基准模型比较,其实验结果如表2表示。从表2中可以看出Linear+HMM在4个指标中效果最差,并且随着数据保持率的下降,其模型的效果显著下降。在25%保持率下,Linear+HMM模型的Recall值与其他基准模型相差不足5%,然而在6.25%中,与其他基准模型相差到16%左右。由此可见,Linear+HMM在低采集率下的轨迹重建效果不理想。

表2 轨迹重建的实验结果
Table 2 Results of trajectory recovery

模型	数据保持率 6.25%				数据保持率 12.5%				数据保持率 25%			
	Recall	Precision	MAE	RMSE	Recall	Precision	MAE	RMSE	Recall	Precision	MAE	RMSE
Linear+HMM	0.286 0	0.278 3	691.389 7	929.420 3	0.371 8	0.363 8	594.613 0	848.258 4	0.480 2	0.472 4	589.486 2	833.419 5
Deep-Move+Rule	0.447 8	0.674 5	470.196 2	883.464 2	0.483 9	0.694 5	414.207 7	834.723 1	0.521 0	0.713 7	367.970 4	804.241 8
T2vec+Decoder	0.451 1	0.689 3	450.044 7	862.796 5	0.491 4	0.706 4	390.363 7	833.424 7	0.584 8	0.714 3	286.995 4	737.889 8
MtrajRec	0.501 5	0.694 4	383.783 2	803.187 9	0.543 5	0.705 0	333.379 4	778.314 5	0.590 4	0.720 4	289.246 0	750.886 5
ZTrajRec-noAttn	0.312 3	0.575 2	742.309 8	1194.758 5	0.379 1	0.596 1	573.715 5	1030.719 4	0.455 1	0.622 5	438.399 1	908.399 4
ZTrajRec	0.532 5	0.704 7	363.523 0	786.788 0	0.578 9	0.715 0	317.203 9	770.119 4	0.631 3	0.738 6	269.701 3	736.640 4

本文模型ZTrajRec通过多头注意力机制捕获轨迹间的长距离依赖关系,并在解码器部分利用注意力来比对当前要还原的轨迹与历史轨迹关系,通过解码器中的LSTM模型记录已还原轨迹路段以及当前路段中的行驶距离信息,准确对下一轨迹点进行还原,从而在性能上有显著的提高。为进一步分析模型对于轨迹重建的效果,解码器中对历史轨迹的注意力模块的权重分布作进一步分析。图2为对应注意力权重图。首先从全局可以看出,整体注意力权重呈现阶梯状。这一现象表明,每个还原的轨迹点都与其邻近点相关。例如,在还原第0点时,注意力只关注对0、1轨迹点,而忽略后面的轨迹点。然

后在还原到第14点时,模型对前1、2、3点权重几乎为0,对中间4、5轨迹点比重相对较大。然而在所有轨迹点重建中,所有点都对起始点分配了较大的权重,这可能是因为低采集率的轨迹重建中只有起始点和终点是明确的。

从表2实验结果还可以看到,ZTrajRec-noAttn模型实验结果相较于ZTrajRec模型,在不同轨迹保持率下各个指标的性能都有大幅度下降,例如在6.25%数据保持率下,MAE、RMSE指标不如线性插补所达到的效果。此实验结果验证了注意力模块的有效性,结合图2注意力权重图也侧面反应了对于轨迹重建情况下,历史轨迹对于当前轨迹点的还原具有不同程度的影响。

3.5 注意力参数调整和效率分析

前文消融实验证明了注意力模块的重要性。本节通过调节注意力的多头个数,来验证对模型整体的影响效果。实验中注意力多头个数选取为{2, 4, 8},分别用A,B,C表示,图3是与表2实验结果对应的4个评价指标结果。从图3中可以看出,多头个数对模型的重建效果的影响并不是很明显。考虑到多头个数的增加可增强模型对不同位置注意力的专注能力,但需要更多的资源去训练,故此本模型选取4作为多头的个数。

除了对比各模型对轨迹重建的准确率外,本节展示各模型在不同保持率下完成轨迹重建所需的推理时间。如图4所示,实验共选取500条轨迹进行重建,其中采用线性插补方式耗时最长,不仅所需时间远远超过其他模型,而且随着保持率增加的情况,推理时间明显可以看到上升趋势。这是因为尽管随着保持率的增加,需要还原的轨迹点变少,但是需要有更多点进行路网匹配算法,而其他基准模型所需时间基本相同。随着保持率的上升,ZTrajRec模型因需要还原的轨迹点变少,其推理所需时间逐渐下降。尽管ZTrajRec所需的推理时间并不是最少的,但是在还原的准确率上要高于其他基准模型。从效率分析的角度上,本次实验证明了本文提出模型的可行性。

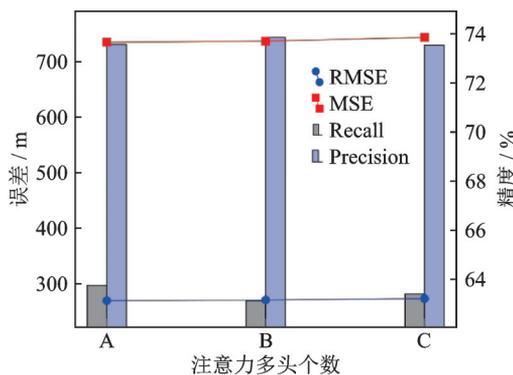


图3 不同多头超参数下ZTrajRec模型的性能指标对比
Fig.3 Performance comparison of ZTrajRec under different multihead hyper-parameters

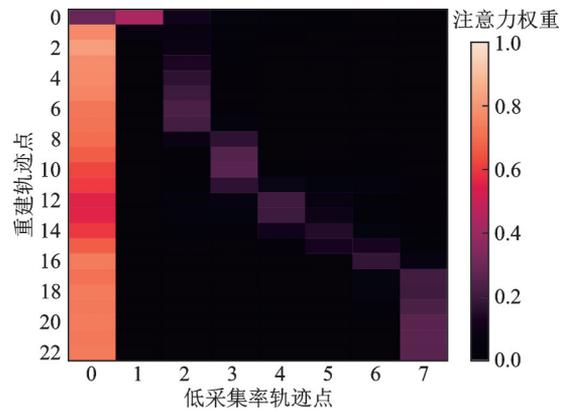


图2 注意力权重图

Fig.2 Attention weight map

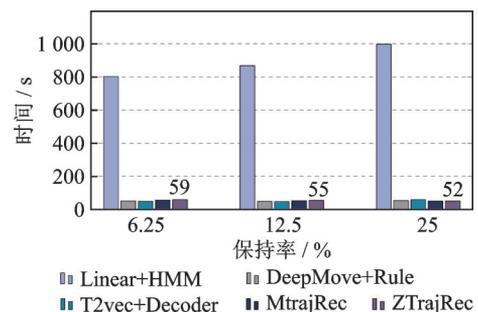


图4 不同保持率下的模型性能指标对比

Fig.4 Performance comparison of models under different keep ratios

3.6 案例分析

本节给出了本文模型 ZTrajRec 对真实轨迹的重建,以进一步进行可视化分析。图 5 展示的是轨迹数据保持率为 25% 的轨迹重建结果,其中黑色圆圈代表低采集率的轨迹数据,蓝色圆圈代表真实的高采集率的路网匹配的轨迹数据,红色圆圈则是模型重建低采集率的路网匹配的轨迹数据。从图 5 中可以看出,低采集率的轨迹数据(GPS 轨迹数据)具有一定的误差,并未在路网上。而经过模型重建的数据则都是还原到路网上,无需后续路网匹配算法。首先,轨迹起点从左下角出发,一直到右上角结束,整条轨迹共发生了 5 次车辆方向行驶的转变。从黑色轨迹点可见,因设备的误差,GPS 轨迹点并未在路网上,但还原的轨迹点依旧可以针对相应的路网,还原出相应的轨迹点。其次,重建的轨迹数据可以正确地反映出真实轨迹的路段,即能够正确规划出租车行走的线路。最后,重建后的轨迹尽管没有完全还原在真实的轨迹点,但大部分重建后的轨迹依旧在真实轨迹附近处,以此能够反映出出租车的真实轨迹情况。

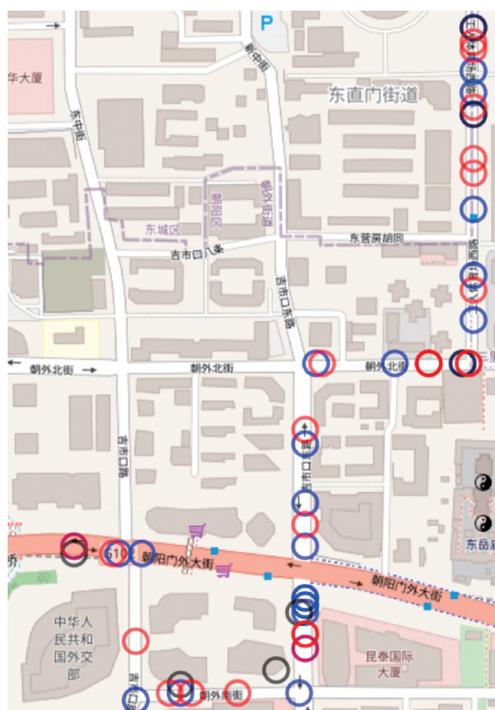


图 5 轨迹重建案例

Fig.5 A case study for trajectory recovery

4 结束语

针对低采集率轨迹数据的重建,本文基于Transformer提出了ZTrajRec轨迹重建模型,并通过历史轨迹对轨迹重建的重要程度,利用多头注意力机制提高模型的有效性。通过实验对真实的北京出租车数据对提出的模型架构进行了验证。虽然本模型相对与基准模型在轨迹重建上有了一定的提升,但本文模型并未对路网结构数据进行充分利用。未来的工作会考虑嵌入更多轨迹特征信息,进一步提升对复杂轨迹的重建能力。

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] ZHANG H, WU H, SUN W, et al. Deepravel: A neural network based travel time estimation model with auxiliary supervision[C]//Proceedings of the 27th International Joint Conference on Artificial Intelligence. Stockholm, Sweden: AAAI Press, 2018: 3655-3661.
- [3] YUAN H, LI G, BAO Z, et al. Effective travel time estimation: When historical trajectories over road networks matter[C]//Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. Portland OR USA: ACM, 2020: 2135-2149.
- [4] SHEN Y, JIN C, HUA J, et al. TTPNet: A neural network for travel time prediction based on tensor decomposition and graph embedding[J]. IEEE Transactions on Knowledge and Data Engineering, 2022, 34(9): 4514-4526.
- [5] LI X, ZHAO K, CONG G, et al. Deep representation learning for trajectory similarity computation[C]//Proceedings of 2018 IEEE 34th International Conference on Data Engineering (ICDE). Paris, France: IEEE, 2018: 617-628.
- [6] HAN P, WANG J, YAO D, et al. A graph-based approach for trajectory similarity computation in spatial networks[C]//Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. New York, USA: Association

- for Computing Machinery, 2021: 556-564.
- [7] HUI B, YAN D, CHEN H, et al. TrajNet: A trajectory-based deep learning model for traffic prediction[C]//Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. New York, USA: Association for Computing Machinery, 2021: 716-724.
- [8] LI M, TONG P, LI M, et al. Traffic flow prediction with vehicle trajectories[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(1): 294-302.
- [9] ZHENG C, FAN X, WANG C, et al. GMAN: A graph multi-attention network for traffic prediction[C]//Proceedings of the AAAI Conference on Artificial Intelligence. [S.l.]: AAAI, 2020: 1234-1241.
- [10] DONG W, LI J, YAO R, et al. Characterizing driving styles with deep learning[EB/OL].(2016-10-08)[2023-10-01]. <https://arxiv.org/abs/1607.03611>.
- [11] REN H, RUAN S, LI Y, et al. MTrajRec: Map-constrained trajectory recovery via Seq2Seq multi-task learning[C]//Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. Singapore: ACM, 2021: 1410-1419.
- [12] WANG S, BAO Z, CULPEPPER J S, et al. A survey on trajectory data management, analytics, and learning[J]. ACM Computing Surveys, 2021, 54(2): 39.
- [13] YUAN J, ZHENG Y, ZHANG C, et al. An interactive-voting based map matching algorithm[C]//Proceedings of 2010 Eleventh International Conference on Mobile Data Management. Kansas City, MO, USA: [s.n.], 2010: 43-52.
- [14] HOTEIT S, SECCI S, SOBOLEVSKY S, et al. Estimating human trajectories and hotspots through mobile phone data[J]. Computer Networks, 2014, 64: 296-307.
- [15] WANG J, WU N, LU X, et al. Deep trajectory recovery with fine-grained calibration using Kalman filter[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 33(3): 921-934.
- [16] NEWSON P, KRUMM J. Hidden Markov map matching through noise and sparseness[C]//Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Seattle Washington, USA: ACM, 2009: 336-343.
- [17] BACHLECHNER T, MAJUMDER B P, MAO H, et al. ReZero is all you need: Fast convergence at large depth[C]//Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence. [S.l.]: PMLR, 2021: 1352-1361.
- [18] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proceedings of Advances in Neural Information Processing Systems. [S.l.]: Curran Associates Inc., 2017.
- [19] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C]//Proceedings of Advances in Neural Information Processing Systems. [S.l.]: Curran Associates Inc., 2014.
- [20] CAO Z, WANG S, SUN D, et al. ITAR: A method for indoor RFID trajectory automatic recovery[C]//Proceedings of Collaborative Computing: Networking, Applications and Worksharing. Cham: Springer Nature Switzerland, 2022: 400-418.
- [21] KINGMA D P, BA J. Adam: A method for stochastic optimization[EB/OL].(2017-01-29)[2023-10-01]. <https://arxiv.org/abs/1412.6980>.
- [22] FENG J, LI Y, ZHANG C, et al. DeepMove: Predicting human mobility with attentional recurrent networks[C]//Proceedings of the 2018 World Wide Web Conference on World Wide Web- WWW' 18. Lyon, France: ACM Press, 2018: 1459-1468.

作者简介:



梅宇生(1995-),通信作者,男,硕士研究生,研究方向:大数据分析,E-mail: meiyusheng246@163.com。



赵卓峰(1977-),男,研究员,研究方向:云计算、海量感知数据处理、服务计算、智慧城市。

(编辑:刘彦东)