

一种车载端为主的城市路网当前与未来速度查询方法

韩京宇* 王彦之 陈进 晏鑫鑫 张怡婷

(南京邮电大学计算机学院 南京 210003)

摘要: 城市智能交通管理中经常查询路段的当前和未来交通速度, 该文提出一种车载边缘为主(VEE)的城市路段速度查询和预测方法: 车载端在速度低于一定阈值时, 与其它车载端交换收集到的数据, 并在本地构建轻量级的当前和历史速度索引, 以支持当前速度查询。为了用尽可能少的模型支持速度预测, 提出根据路段拓扑同构将路网划分成若干路段等价类, 根据周期性时窗和路段等价类将整个时空划分成若干模型等价类, 同一个模型等价类的路段在给定时窗呈现相似的交通运行模式。针对每个模型等价类, 车载端和数据中心配合进行联邦学习, 训练长短期记忆模型(LSTMs)并存储在车载端, 以响应车载端对附近未来交通状况的查询。每个车载端本地索引数据、本地响应查询, 避免了查询响应延迟和通信拥塞; 数据保存在车载端, 而非集中存放, 避免了安全攻击导致的隐私泄露。

关键词: 智能交通; 查询; 预测; 索引; 联邦学习

中图分类号: TN929.53; TP311.132

文献标识码: A

文章编号: 1009-5896(2024)09-3722-09

DOI: 10.11999/JEIT240102

A Mobile-Side-Dominant Method for Querying Present and Future Velocity on Urban Roads

HAN Jingyu WANG Yanzhi CHEN Jin YAN Xinxin ZHANG Yiting

(School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: Querying present and future traffic velocities of road segments is a routine task in urban intelligence transportation management, and a Vehicle-equipped-Edge Dominant (VEE) method is proposed to answer the querying of present and future velocity of urban road segments. The collected data is exchanged with the other mobile sides by every vehicle-equipped mobile side when the mobile side's speed falls below a given threshold, and the light-weighted present and history velocity indexes are constructed locally to support the querying of present velocity. To train as few models as possible to predict future velocities, a road network is proposed to be partitioned into a set of road-segment clusters based on the segments' topological morphism and the spatio-temporal space is proposed to be partitioned into a set of model-equivalence classes according to the periodic time windows and road-segment clusters. The similar traffic patterns are exhibited by the road segments in the same model-equivalence class within the given time window. For every model-equivalence class, the federated learning is performed between the mobile sides and the data center to train the Long Short-Term Memories (LSTMs) which are stored at the mobile sides to answer the querying of future velocities of nearby areas. Data is indexed by every mobile side and queries are answered locally, thus the query response latency and possible communication congestion can be avoided. Further, data is stored at the mobile sides, rather than at one data center, so as to prevent the privacy leakage due to security attacks.

Key words: Intelligence transportation; Query; Prediction; Index; Federated learning

1 引言

随着城市人口的增多和私家车的普及, 城市交

通愈发拥挤。同时, 车载端的存储、计算和通信能力不断提高, 不仅可以感知和传输数据, 而且可以进行本地计算。车载端的智能导航、交通调度和预警依赖的核心操作是查询路段的当前和未来速度。为实现速度查询, 通常的软件架构依赖数据中心, 车载端不时地将位置、时间戳、速度等汇报给数据中心, 数据中心建立索引, 以响应车载端对当前和未来的交通速度查询^[1]。根据数据来源, 支持当前

收稿日期: 2024-02-26; 改回日期: 2024-06-15; 网络出版: 2024-06-24

*通信作者: 韩京宇 jyhan@njupt.edu.cn

基金项目: 江苏省重点研发计划(BE2022065-5)

Foundation Item: Jiangsu Provincial Key Research & Development Plan (BE2022065-5)

速度查询的方法分成基于单源数据和基于融合数据两类。基于单源数据的方法仅根据1种类型的数据回答当前速度查询^[2-4]。基于融合数据的方法集成多源数据回答当前速度查询^[5]。预测未来速度的方法分成浅层模型^[6]、卷积神经网络^[7,8]和图神经网络^[9,10]3类。但这些方法需要将车载端感知的数据传输到数据中心,不仅消耗网络带宽,而且查询集中在数据中心处理,响应会有延迟。此外,数据集中存放,容易受到安全攻击,导致隐私泄露^[11]。

随着无线通信技术的进步和车载设备存储与计算能力的提高,边缘计算飞速兴起。边缘计算通常把数据和计算布置在数据生成位置附近^[12,13],多采用云-边-端协同方案^[13,14]。也有少量主要依赖车载端提供服务的,如文献^[12]的Trellis利用车辆的Wi-Fi信号探测移动设备,进而对车内和车外人员的流动性进行捕捉,提供起点、终点、行人活动趋势等分析,但Trellis框架中车辆运行时不与其他车辆交换数据,也没提供处理当前和未来查询的具体方法。

综上所述,设计一种主要依靠车载端的本地存储和计算回答当前和未来速度查询,避免数据集中存放,实现低延迟的本地响应,成为一种紧迫需求。注意到,如果查询的目标时空距离车载端的当前时空很近,可以根据最近收集到的周边速度数据响应查询;如果查询的目标时空和车载端的当前时空比较远,则可以根据车载端积累的历史数据进行推断,实现查询的本地响应。为此,本文提出一种城市路网上车载边缘为主(Vehicle-equipped-Edge Dominant, VED)的方法响应当前和未来速度查询。当车载端的速度降低到某一阈值时,可以根据一定的隐私保护协议与相遇的车载端进行数据交换^[15,16],共享其它路段的最新交通状况。如图1,进入彼此的通信范围时,车载端A与反方向的车载端B通过长期演进车用无线(Long Term Evolution - Vehicle to Everything, LTE-V2X)或下一代车用无线(Next generation Radio - Vehicle to Everything, NR-V2X)等通信链接交换数据^[17],获得前方路段的交通信息;A可以从车载端C获得垂直方向路段的交通状况。每个车载端索引收集到的数据,并存储速度预测模型,以回答当前和未来速度的查询。随着时间的推移,车载端积累的数据越多,查询结果也越精确。

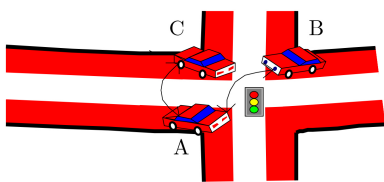


图1 路网上车载端数据交换的示例

VED方法主要包括数据收集、当前查询响应和未来查询响应3个环节。车载端在速度低于阈值或停留在红灯路口时,进行数据广播,收到广播的车载端根据数据新鲜度和样本的路段重要性决定保留哪些数据。为了回答当前速度查询,车载端构建当前速度索引(Present Velocity Index, PVI)和历史速度索引(History Velocity Index, HVI)。前者对最近时间的数据进行索引,后者对历史数据进行索引。查询当前速度时,在PVI中搜索目标时空单元。如果目标时空单元没有数据,使用矩阵补全方法根据PVI和HVI推断当前交通状况。为了支持未来速度查询,提出根据路段同构性将路段划分成若干路段等价类,然后根据交通流峰谷时窗和路段等价类将整个时空划分成若干模型等价类。针对每个等价类,车载端与数据中心采用联邦学习,协同训练一组长短期记忆(Long Short-Term Memory, LSTM)模型^[18]。训练好的模型保存在车载端,以支持对未来速度的查询。本文主要贡献如下。(1)针对城市路网的交通速度查询,提出一种车载端为主的解决方案,车载端本地存储数据,本地响应查询。(2)根据Z曲线^[19],设计轻量级的当前速度索引和历史速度索引,在有限的存储预算下,支持车载端的当前速度查询。(3)提出路段等价类的具体划分方法,减少速度预测模型的数量、提高模型训练效率,支持未来速度查询。

2 基本概念与数据收集

定义1 (有向路网和跳距)有向路网DN由节点 J 和有向路段RS组成,记为 $DN=(J,RS)$ 。两个节点间的有向路段 $rs \in RS$ 对应一个4元组 $(nl, len, din, dout)$,其中 nl 为车道数, len 为路段长度, din 为进入 rs 的路段数, $dout$ 为从 rs 引出的路段数。给定源路段 rs^1 与目标路段 rs^2 ,跳距 $hd(rs^1, rs^2)$ 是从 rs^1 到 rs^2 的最短路径上经过的节点个数。

车载端在有向路网上以天为周期运动,每个周期在路段-时隙网格中建模:

定义2 (路段-时隙网格)给定时隙阈值 τ ,网格SG由有向路段RS和时隙TS决定,记为 $SG=(RS, TS)$, $rs \in RS$ 和 $ts \in TS$ 分别是路段和时隙序号。 ts 对应起始时间戳 $start$ 和终止时间戳 end ,满足 $end-start = \tau$ 。

在提出的VED方法中,车载端在SG中收集速度样本,回答当前和未来速度查询。当前速度查询 $pq=(rs,ts)$ 询问路段 rs 在时隙 ts 的平均速度,其中 $0 < now.ts - ts < ck$, $now.ts$ 为查询发出的时隙序号, ts 为目标时隙序号, ck 为时隙差阈值。未来速度查询 $fq(rs, ts)$ 询问路段 rs 在未来时隙 ts 的平均速度,

$ts - \text{now} \cdot ts = 1$, 其中 ts 是目标时隙序号。为了在相遇的车辆间传播速度样本, 车辆在运行速度低于速度阈值或在红灯前停留时, 通过LTE-V2X或NR-V2X链路广播其收集到的速度样本。

定义3 (速度样本)速度样本 ps 是一个5元组 $(rs, ts, dd, \text{start}, \text{end})$, 表示车辆在 $[\text{start}, \text{end}]$ 时间在路段 rs 上行驶 dd 的距离, 其中 ts 是时隙序号。

邻近的车载端接收到广播时, 根据样本新鲜度和路段重要性确定样本交换优先级。令 $\gamma = \text{now} - (\text{ps}.\text{start} + \text{ps}.\text{end})/2$, 样本 ps 的新鲜度为

$$fn(ps) = \begin{cases} 1 - \frac{\gamma}{C}, & \gamma \leq T \\ 0, & \gamma > T \end{cases} \quad (1)$$

其中, C 是使 $0 < fn(ps) \leq 1$ 的正则化因子, T 是失效时长阈值。不同类型的路段在路网中发挥不同的作用, 因此要同时考虑样本 ps 对应路段的重要性, 从进入路段数、引出路段数, 车道数以及路段长度来衡量, 定义为

$$si(ps) = \frac{(\text{ps}.\text{rs}.\text{din} + \text{ps}.\text{rs}.\text{dout}) \times \text{ps}.\text{rs}.\text{nl} \times \text{ps}.\text{rs}.\text{len}}{(\text{din}^{\max} + \text{dout}^{\max}) \times \text{nl}^{\max} \times \text{len}^{\max}} \quad (2)$$

其中, $\text{ps}.\text{rs}.\text{din}$ 和 $\text{ps}.\text{rs}.\text{dout}$ 分别代表 $ps.\text{rs}$ 的进入和引出路段数; $\text{ps}.\text{rs}.\text{nl}$ 和 $\text{ps}.\text{rs}.\text{len}$ 分别代表 $ps.\text{rs}$ 的车道数和长度; din^{\max} , dout^{\max} , nl^{\max} 和 len^{\max} 代表进

$$dt^{(h+1)} = dt^{(h)} + \text{ps}_{h+1}.\text{dd}, tm^{(h+1)} = tm^{(h)} + (\text{ps}_{h+1}.\text{end} - \text{ps}_{h+1}.\text{start}), num^{(h+1)} = num^{(h)} + 1 \quad (3)$$

$$\sigma^{(h+1)} = \frac{h \times (u^{(h)} - u^{(h+1)})^2 + \left(\frac{\text{ps}_{h+1}.\text{dt}}{\text{ps}_{h+1}.\text{end} - \text{ps}_{h+1}.\text{start}} - u^{(h+1)} \right)^2 + h \times (\sigma^{(h)})^2}{h + 1} \quad (4)$$

每个时隙结束时, 对应时空单元的样本概况被合并到HVI中, 同时刷新PVI的新时隙单元格。

3.2 HVI维护

交通流每天周期性地变化, HVI记录每个时空单元的历史平均速度分布。查询当前速度时, 如果PVI中缺少当前时隙数据, 则根据历史数据推断当前速度。HVI中每个时空单元 $SG(rs, ts)$ 的速度模式用截尾正态分布 $vp(u, \sigma)$ 拟合, 表示在时隙 ts 路段 rs 上, 速度围绕 u 以标准差 σ 变化。设车载端原来的速度模式 $vp_1(u_1, \sigma_1)$, 新接收的数据对应的速度模式是 $vp_2(u_2, \sigma_2)$, 则合并后的速度模式 $vp_{12}(u_{12}, \sigma_{12})$ 为

$$u_{12} = \frac{u_1\sigma_1^2 + u_2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}, \sigma_{12} = \sqrt{\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}} \quad (5)$$

HVI中叶节点的各项记录了到目前形成的若干个代表性速度模式, 每个代表性速度模式 (vp, num)

入路段数、引出路段数、车道数和路段长度的最大值。综上所述, ps 的交换优先级定义为 $ep(ps) = fn(ps) \cdot si(ps)$, $ep(ps)$ 越大, ps 越有可能被转发到其它车载端。每个车载端根据交换优先级阈值 η 维护一个优先样本队列。

3 处理当前速度查询

为了回答对邻近区域当前速度的查询, 每个车载端构建 B^+ 树, 索引当前速度样本和历史速度样本。当前速度索引记录各路段上当前时隙的速度信息, 历史速度索引记录各个路段、时隙在过去各天的平均速度分布。给定2维的路段序号 rs_i 与时隙序号 ts_j , 车载端计算1维的Z曲线序号^[19], 从而将对应的时空单元速度样本插入 B^+ 树索引。

3.1 PVI的数据结构与维护

设一个车载端收集了 h 个属于 $SG[rs, ts]$ 的样本, 用样本概况对它们刻画如下:

定义4 (样本概况)给定 h 个数据样本 $\{ps_1, \dots, ps_i, \dots, ps_h\}$, 其样本概况是一个4元组 $sk=(dt, tm, num, \sigma)$, 其中 $dt = \sum_{i=1}^h ps_i.\text{dd}$ 是所有样本的移动距离之和。 $tm = \sum_{i=1}^h (\text{ps}_i.\text{end} - \text{ps}_i.\text{start})$ 为所有样本运行时间之和, $num = h$ 为样本个数, σ 是平均速度 $u = dt/tm$ 的标准差。车载端收到一个样本 ps_{h+1} 时, sk 的4个值更新为

中 vp 和 num 分别是速度模式和样本数。每过一个时隙, PVI中时空单元的样本概况与HVI中相应单元的速度模式进行合并: 首先, 计算PVI中 $SG[rs, ts]$ 对应的速度模式和HVI中各个速度模式的Jensen-Shannon散度 JSD ^[20], 寻找 JSD 最小的目标模式。如果最小 JSD 小于给定的距离阈值, 则直接合并。否则分两种情况处理: 如果最小 JSD 值大于给定的距离阈值, 且该叶结点速度模式的数量小于设定阈值, 则在HVI的 $SG[rs, ts]$ 中插入一个新的速度模式。否则, 根据式(5)进行速度模式合并。

3.3 回答当前交通速度查询

车载端 ob 发起对邻近路段的当前速度查询 $pq=(rs, ts)$, 按照如下过程求解: 首先, 在PVI中搜索目标单元 $SG[rs, ts]$, 如果PVI中记录了目标单元格的样本概况, 直接根据样本概况计算平均速度。否则, 如果HVI中对应单元格有数据, 则根据存储的速度分布计算平均速度。若PVI和HVI都没

有对应的数据，计算 $SG(rs,ts)$ 对应的恢复矩阵 AV ，用矩阵的对应元素代表目标速度。

4 处理未来速度查询

车载端和数据中心协同，使用联邦学习^[21]训练LSTM模型，每个车载端保存训练好的模型以响应未来速度查询 $f_q=(rs,ts)$ ，为此提出根据 mnk 时空立方的数据预测 $SG(rs,ts)$ 的速度。

定义5 (mnk 时空立方) 给定时空单元 $SG(rs,ts)$ ，它的 mnk 时空立方 $mnk-SH(rs,ts)$ 由 mk 进入单元 $mk-IC(rs,ts)$ 和 nk 引出单元 $nk-OC(rs,ts)$ 组成。每个进入单元 $ic \in mk-IC(rs,ts)$ 满足 $0 \leq hd(ic,rs,rs) \leq m$ 和 $0 \leq dif(ic,ts,ts) < k$ ，其中 $dif(ic,ts,ts) = ts - ic.ts$ 是两个时隙序号之差。同理， $nk-OC(rs,ts)$ 代表 $SG(rs,ts)$ 的引出单元满足 n 和 k 的限制。

图2显示了路段 z 的1层引出与2层进入路段， $2-1-3-SH(z,5)$ 时空立方由进入单元 $2-3-IC(z,5)$ 和引出单元 $1-3-OC(z,5)$ 构成。图3显示了 $2-3-IC(z,5)$ ，其中 $k=3$ 代表第5时隙之前的3个时隙(含第5时隙)， $m=2$ 代表距离 z 路段2个跳距以内的进入路段。用训练好的LSTM模型预测 $SG(rs,ts)$ 的速度时，用 rs 在前 $k-1$ 个时隙中 m 个跳距内的进入路段上的速度和 n 个跳距内的引出路段的速度作为输入。

如果为每个单元 $SG(rs,ts)$ 训练一个LSTM模型，不仅模型多，而且训练数据稀疏，提出构建模型等价类的方法减少模型个数。首先，按路网特性把路段分成 L 个路段等价类。其次，各天的时间维度根据交通流划分成高峰与低谷两个时间段^[7]。结合路段类划分和时间划分，整个时空划分成 $2L$ 个模型等价类。

4.1 路段等价类的划分

路段的交通流不仅受路段等级影响，同时受进入路段和引出路段的影响。提出采用 mn 同构衡量路段相似，依赖于 m 度进入同构和 n 度引出同构。为此，先定义孪生路段：

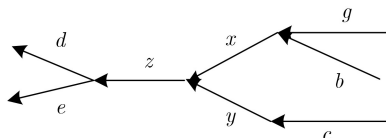


图2 路段 z 的进入与引出路段

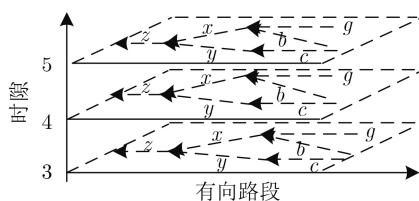


图3 时空立方 $2-1-3-SH(z,5)$ 中的 $2-3-IC(z,5)$

定义6 (孪生路段) 给定车道-长度网格 $LLG(nl,len)$ ，其中 nl 表示车道数，间距是 α ， len 表示路段长度，间距是 β 。给定有向路段 rs^1 和 rs^2 ，如果根据车道数和长度它们落入 LLG 的同一单元格，则 rs^1 和 rs^2 为孪生路段。孪生路段有相近的交通流承载能力和通行速度。

定义7 (孪生双射) 给定两个路段集合 RS^1 与 RS^2 ，若它们间存在一个双射 bh ，使得 $(rs^1, rs^2) \in bh(rs^1 \in RS^1, rs^2 \in RS^2)$ 是一对孪生路段，则 bh 是孪生双射关系，记为 $RS^1 \leftrightarrow RS^2$ 。一对孪生路段 rs^1 与 rs^2 显然满足 $rs^1 \leftrightarrow rs^2$ ，同时认定 rs^1 与 rs^2 满足0度进入同构和0度引出同构。在此基础上， mn 同构定义如下：

定义8 (mn 同构) 给定两个路段 rs^1 与 rs^2 ，令 BS^1, BS^2 分别代表直接进入 rs^1 和 rs^2 的路段集合。若满足以下两点：(1) rs^1 和 rs^2 满足孪生双射；(2) BS^1 和 BS^2 满足孪生双射 bh ，并且对任意的 $bh(u)=v (u \in BS^1, v \in BS^2)$ ， u 和 v 是 $m-1$ 度进入同构，则称 rs^1 和 rs^2 为 m 度进入同构。同理可得 n 度引出同构。如果 rs^1 与 rs^2 既满足 m 度进入同构，又满足 n 度引出同构，则称 rs^1 与 rs^2 满足 mn 同构。

图4演示了图5中 z 和 z' 两个路段不同阶段的进入同构判定。阶段1， z 和 z' 是0度进入同构。阶段2，由于 x 和 x' 、 y 和 y' 是0度进入同构的，推导出 z 和 z' 是1度进入同构。阶段3，由于 g 和 g' 、 b 和 b' 、 c 和 c' 是0度进入同构，则 x 和 x' 、 y 和 y' 是1度进入同构的，进而推导出 z 和 z' 是2度进入同构。

为了判定DN中任意两个路段是否 mn 同构，将路段的每个 m 跳进入路径和 n 跳引出路径转换为一个特征符序列，每个特征符对应 $LLG(nl, len)$ 中的一个单元格。这样，每个路段就对应一个特征符序

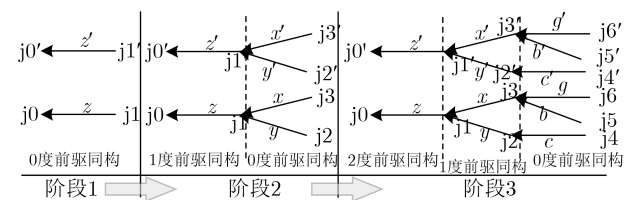


图4 进入路段同构的推导示例

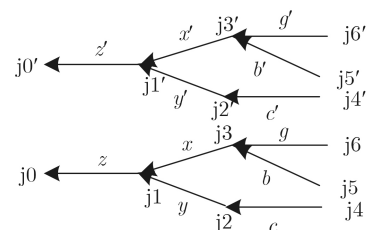


图5 z 和 z' 的进入路段示例

列的集合, 哈希后, 落入同一个桶的路段则是 mn 同构的。整个过程如算法1所示: 首先以深度优先的顺序遍历DN中的全部有向路段, 对于每个路段, 获取其 m 跳的进入路段和 n 跳的引出路段。inSphere(\cdot) (算法2) 描述了如何提取 m 跳进入路段; outSphere(\cdot) 用来提取 n 跳引出路段; featureCode(\cdot) 将所有序列拼接成的特征字符串, 不再叙述细节。最后, 根据特征字符串映射到对应的桶, 每一个桶中的路段具有相同的 mn 同构性。

算法1的时间复杂度主要取决于处理所有路段的 m 度进入路段和 n 度引出路段的时间。设 W 表示路段的 m 度进入路段的最大数目, Q 表示路段的 n 度引出路段最大数目, 则算法2在最坏情况下的时间复杂度是 $O(|DN.RS| \cdot (m \cdot W + n \cdot Q))$ 。

算法1 同构路段划分

输入: 路网DN, 同构参数 m 和 n , 特征标签集合 L
输出: 路段的同构分组

- (1) 将DN的第1个路段sg压入栈SK中
- (2) while SK非空 do
- (3) sg \leftarrow SK.pop(\cdot)
- (4) if sg未被处理过 then
- (5) visit (sg)
- (6) sg的前驱路段insgs \leftarrow inSphere (DN, sg, m)
- (7) sg的后继路段outsgs \leftarrow outSphere (DN, sg, n)
- (8) hc \leftarrow featureCode (insgs, outsgs, L)
- (9) 根据hc将sg归类
- (10) end
- (11) if sg的邻接路段 w 未被分类过 then
- (12) push(w)
- (13) end
- (14) else
- (15) sg = SK.pop(\cdot)
- (16) end

算法2 inSphere

输入: 路网DN, 进入同构参数 m , 目标路段tsg
输出: 路段tsg的 m 跳距内的进入路段

- (1) 初始化队列 Q 并将tsg入队
- (2) visit (tsg), ret \leftarrow \emptyset
- (3) while Q 非空 do
- (4) sg \leftarrow Q .pop(\cdot), ret \leftarrow ret \cup {sg}
- (5) if |sg| $<$ m then
- (6) 获取sg的所有直接前驱并将它们送入队列 Q
- (7) end
- (8) end

4.2 联邦学习训练LSTM模型

LSTM模型的每个训练样本是 $1, 2, \dots, k-1$ 时隙的速度和第 k 个时隙目标路段的速度, 其中前 $k-1$ 个时隙的每个时隙对应一组 mn 空间单元的速度。由于速度样本都存储在各个车载端, LSTM模型的训练由车载端与数据中心合作, 包含车载端选择和联邦学习训练两个阶段构成。

针对每个模型类, 数据中心广播一个4元组 (m, n, k, lt) , 其中 m, n, k 是 mnk 时空立方参数, lt 是目标时空单元列表。当车载端ob收到请求后, 搜索PVI与HVI, 以找到符合条件的时空单元QC(ob) = $\{c_1, \dots, c_i, \dots, c_N\}$, 其中 $c_i = (rs, ts) \in lt$, 并且 c_i 有数据。进而计算NM(ob)和NS(ob)两个指标。NM(ob)是QC覆盖的 mnk 时空立方中的非空单元数

$$NM(ob) = \sum_{c_i \in QC(ob)} \sum_{cl \in mnk-SH(c_i, rs, c_i, ts)} \pi(cl) \quad (6)$$

其中, $\pi(cl)$ 是 cl 所对应的 mnk 时空立方中的一个单元, 若 cl 有数据, 则 $\pi(cl)$ 为1, 否则为0。进而计算可用数据样本总数, 记为NS(ob)

$$NS(ob) = \sum_{c_i \in QC(ob)} \prod_{cl \in mnk-SH(c_i, rs, c_i, ts)} |cl| \quad (7)$$

其中, $|cl|$ 是ob拥有的 cl 单元中的样本数。随后, QC(ob), NM(ob)以及NS(ob)会被发送到数据中心。设数据中心获得了 w 个车载端的响应, 记为 $AO = \{ob_1, \dots, ob_i, \dots, ob_w\}$, 要选取AO的一个子集SO。为此, 定义单元格覆盖率 f^c 和样本覆盖率 f^s

$$f^c = \frac{\sum_{ob \in SO} NM(ob)}{|lt| \cdot w}, f^s = \frac{\sum_{ob \in SO} NS(ob)}{\sum_{ob \in AO} NS(ob)} \quad (8)$$

定义单元格重叠率

$$f^o = \frac{\sum_{c_i \in lt} \text{sgn}(SO中两个以上ob有非空 c_i)}{|lt|} \quad (9)$$

其中, $\text{sgn}(\cdot)$ 是符号函数。数据中心通过求解最大化 f^c 和 f^s 、最小化 f^o 的多目标优化确定参加训练的移动端。

当参与训练的车载端确定后, 数据中心和车载端采用联邦学习的参数平均策略^[21], 以较少的轮数训练出高精度的LSTM模型。对于每轮训练, 数据中心随机选择一定比例的车载端, 每个车载端根据梯度下降更新LSTM模型, 数据中心对各车载端的LSTM模型参数进行平均。

4.3 响应未来速度查询

LSTM模型训练好之后被保存在车载端。给定

一个未来速度查询 $f_q=(rs,ts)$ ，其中 ts 是一个未来时隙，车载端调用**算法3**进行回答。为了预测 $SG(rs,ts)$ 的速度，将 $mn(k-1)-SH(rs,ts-1)$ 内的速度模式输入LSTM模型，预测值作为查询值。

5 实验评估

实验在一台32核处理器、32 GB RAM的PC机上运行，程序用Python语言实现(<https://github.com/hjyresearch228/mdm/blob/main/VE D.zip>)，采用了模拟数据集和真实数据集。为生成模拟数据集，从OpenStreetMap中下载了北京市城区的路网数据，使用城市移动模拟器(Simulation of Urban MObility, SUMO)系统(<https://eclipse.dev/sumo/>)仿真7 d内的100 000个车载端，对应100 000条移动轨迹。每条轨迹的起始时间戳在7 d内均匀分布，每隔20 s车载端在速度低于5 m/s时，将采集的速度样本广播给100 m范围内的车载端。真实数据采用Geolife轨迹数据集(<https://www.microsoft.com/en-us/download/details.aspx?id=52367>)。每个用户收集的数据对应一个车载端，由于有些区域轨迹稀疏，选择30个轨迹稠密区域，每个稠密区域对应的车载端数目介于32~85，运行时长介于1 200~36 000 min。在模拟数据集上，每个车载端运行24 h后，每隔5 min交替执行一个当前查询或未来查询。下文用Pr- $s-t$ 表示当前查询的路段和时隙范围，比如Pr-0-0查询当前路段、当前时隙的速度，Pr-2-1查询当前路段的2跳进入或引出路段在前1个时隙的速度。未来查询关注当前路段、未来1个时隙的速度。汇报的实验结果是所有车载端当前查询或未来查询的平均结果。除特殊说

算法3 响应未来交通查询

输入：车载端的LSTM模型，对未来交通的查询 $f_q(rs,ts)$ ，当前速度索引PVI，恢复矩阵 \mathbf{AV}

输出：对应的时空单元 $SG(rs,ts)$ 的预测速度

- (1) $lstm \leftarrow$ 车载端对应于 $SG(rs,ts)$ 的LSTM模型
- (2) $sh \leftarrow$ PVI中对应于 $mn(k-1)-SH(rs,ts-1)$ 时空范围的样本概况
- (3) 预测集 $pv \leftarrow \emptyset$
- (4) for $cl \in sh$ do
- (5) $vv \leftarrow$ 计算 cl 中的速度
- (6) if cl 为空 then
- (7) $vv \leftarrow \mathbf{AV}[rs,ts]$ 中的数据
- (8) end
- (9) 将 vv 加入 pv
- (10) end
- (11) $ret \leftarrow lstm(pv)$
- (12) return ret

明， $mnk=2$ ，即时空立方的3个参数都为2，间隙阈值 $\tau=6$ min，失效时长阈值 $T=36$ min。

5.1 数据采集的评估

对于数据采集，关注数据的新鲜度与时空单元的覆盖率，篇幅所限，只汇报时空单元覆盖率的实验结果。令 M 为路网上车载端的集合，车载端 ob 的时空单元覆盖率按轨迹进行计算。对于一个车载端，它的当前轨迹数据记为 $PT(ob)$ ，历史轨迹数据记为 $HT(ob)$ ，那么按当前轨迹数据 $PT(ob)$ (或历史轨迹数据 $HT(ob)$)在 mnk 时空范围内的时空单元覆盖率记为

$$gc(ob, (m, n, k)) = \frac{\sum_{tr \in PT(ob)} \sum_{(SG[i,j] \cap tr) \neq \emptyset} cnt^{ob}(mnk - SH(i, j))}{\sum_{tr \in PT(ob)} \sum_{(SG[i,j] \cap tr) \neq \emptyset} |mnk - SH(i, j)|} \quad (10)$$

其中， $|mnk - SH(i, j)|$ 中时空单元的个数， $cnt^{ob}(mnk - SH(i, j))$ 是 ob 中 $mnk - SH(i, j)$ 里的非空时空单元。因此，对任一 mnk 时空范围，其覆盖率记为

$$gc(m, n, k) = \frac{\sum_{ob \in M} gc(ob, (m, n, k))}{|M|} \quad (11)$$

时空单元覆盖率描述了最近一段时间邻近区域的数据采集情况。

将 mnk 立方参数设置为2和3，即 $m=2, n=2, k=2$ 和 $m=3, n=3, k=3$ ，将 η 从0.1增加到0.9，**图6**和**图7**分别显示了在模拟数据集和Geolife数据集上 gc 的变化。可以看出，随着 η 的增大，时空覆盖率逐步减小，这是因为随着 η 的增大，车载端间传播的数据相应变少的缘故。

5.2 路段范围和时隙范围对当前速度查询精度的影响

设 Q 为所有车载端处理的当前速度查询集合，当前查询精度定义为

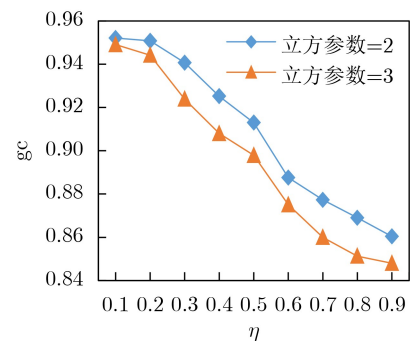


图6 模拟数据集上 gc 随 η 的变化

$$qa(Q) = \frac{\sum_{q \in Q} (1 - MAPE(q))}{|Q|} \quad (12)$$

其中, $MAPE(q)$ 为查询 q 的百分比误差, 记为

$$MAPE(q) = \frac{|SG[q.rs, q.ts] \cdot v^{qv} - SG[q.rs, q.ts] \cdot v^{rl}|}{SG[q.rs, q.ts] \cdot v^{rl}} \quad (13)$$

其中, $SG[q.rs, q.ts] \cdot v^{qv}$ 是根据处理查询 q 的车载端收集的 $SG[q.rs, q.ts]$ 的样本计算的平均速度, $SG[q.rs, q.ts] \cdot v^{rl}$ 是 $SG[q.rs, q.ts]$ 中所有的样本的平均速度。图8显示了固定目标查询路段为当前路段, 时隙范围从0变化到3时, 查询精度在Geolife数据集上的变化。图9显示了固定目标查询时隙为当前时隙, 路段范围从0变化到3时, 查询精度的变化。可见, 随着相对于当前路段-时隙的距离加大、

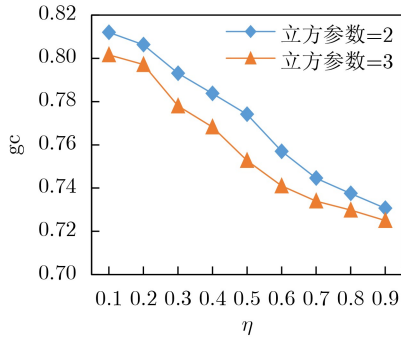


图7 Geolife数据集上gc随η的变化

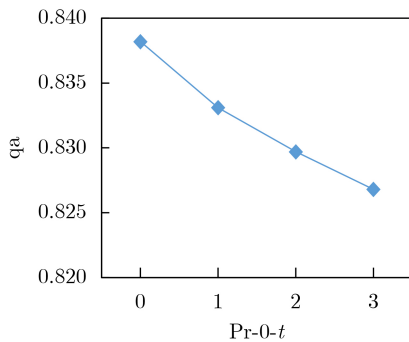


图8 Geolife上时隙范围对当前查询精度的影响(当前路段)

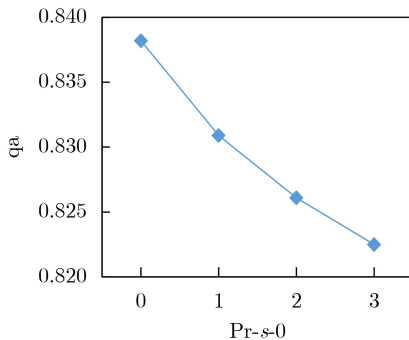


图9 Geolife上路段范围对当前查询精度的影响(当前时隙)

时间远离, qa 会缓慢下降。这是因为, 随着时空目标远离当前时空, 车载端收集到的对应时空单元的速度样本会减少, 但由于积累的历史数据的查询支持, 精度缓慢降低。

5.3 未来速度查询

5.3.1 模型数量的影响因素

设置车道长度网格的 $\alpha = 2, \beta = 500$ m。图10描述了Geolife数据集上将 mn 同构参数从1增加到4时, 模型数量的变化。 mn 越大, 生成的模型越多。这是因为 mn 越大, 落入同一等价类的路段数目越少, 等价类就越多。令 $mn=2, \alpha=2$, 将 β 从100增加到2 000, 图11描述了在Geolife数据集上模型数量的变化。

随着 β 的增大, 模型的数量减少。这是因为 β 越大, 划入同一等价类的路段越多, 等价类就减少。 α 对等价类数目具有和 β 相似的效果, 不再叙述。

5.3.2 mnk参数对未来查询准确率的影响

为了减少随机因素带来的误差, LSTM模型训练采用五折交叉验证。设所有车载端的未来速度查询集合 $FQ = \{fq_1, fq_2, \dots, fq_N\}$, 每个查询 $fq_i(rs, ts)$ 预测路段 rs 在间隙 ts 的速度。未来查询的准确率定义为

$$pa = \frac{1}{N} \sum_{i=1}^N \left(1 - \left| \frac{y_i - py_i}{y_i} \right| \right) \quad (14)$$

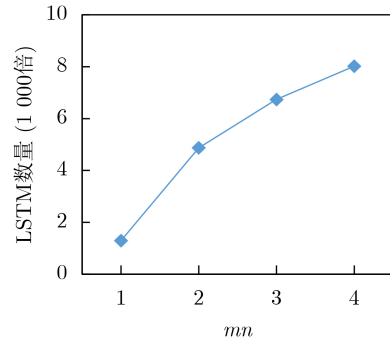


图10 Geolife上LSTM的数量随mn同构值的变化

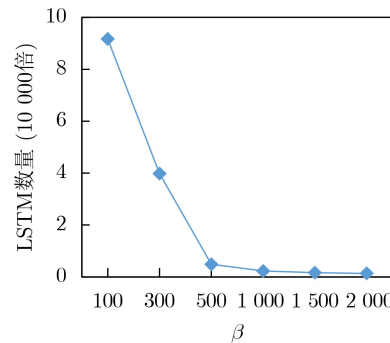


图11 Geolife上LSTM的数量随β的变化

其中, y_i 是实际速度, py_i 是预测的速度。图12描述了当 k 为6时, mn 从1~4时的未来查询准确率。图13描述了当 $m=n=2$ 时, 时隙数 k 从1变化到6时, 未来查询准确率的变化。由于 mnk 越大, 训练LSTM模型时使用的时空信息越多, 因此查询的精度也相应提高。

5.4 对比实验

Trellis^[12]是利用车辆感知交通流的代表性框架, 每个车辆只保留自己感知的数据, 同时把数据发到数据中心进行存储, 车辆相遇时不进行数据的交换。在Geolife数据集上, 令 $T=36$, $\eta=0.7$, 将时空覆盖度量参数 mnk 从1增加到4, 图14显示了VED和Trellis框架的时空单元覆盖率。VED方法的时空覆盖率随着 mnk 的增加略有下降, 而对于Trellis框架, 由于每个车载端只保存自己感知到的数据, 时空覆盖率低于VED, 并且随着度量范围的扩大, 和VED方法的差距会变大。

由于Trellis没有历史数据保存机制, 无法进行未来查询效果比较。图15对比了两种方法的当前查询的精度。如图所示, 随着Pr-s-t中 s 和 t 的同步增加, VED的精度缓慢下降, 因为VED方法中, 车载端可以有效地收集邻近区域的当前数据, 并进行累积。对于Trellis框架的车载端, 只存储自身感知的数据, 因此随着查询范围Pr-r的增加, 精度迅速下降。

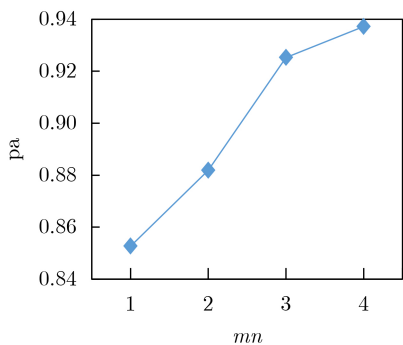


图 12 未来查询精度随 mn 的变化($k=6$)

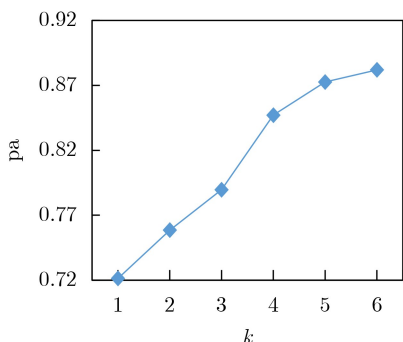


图 13 未来查询精度随时隙数 k 的变化($m=n=2$)

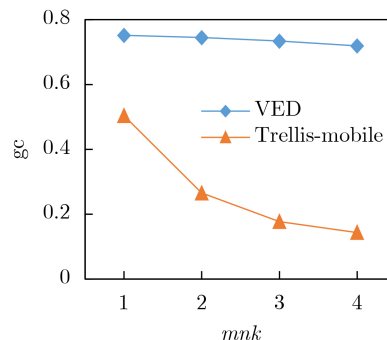


图 14 VED和Trellis时空覆盖率对比

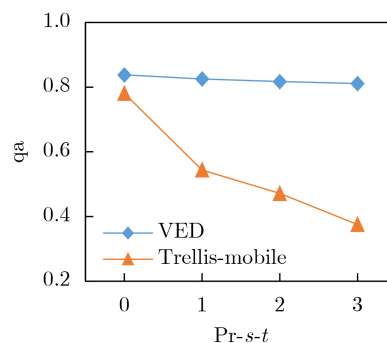


图 15 不同时空范围下当前查询精度的对比

6 结论

城市路网的交通状况影响生产和生活的方方面面。以往的解决方案主要依赖集中式的架构来支持当前和未来速度查询。一方面会有查询响应延迟和可能的服务拥塞, 另外数据集中存放, 容易遭受安全攻击。为此本文提出了一种车载边缘为主的都市路网交通查询和预测方法。首先, 车载端结合数据新鲜度和路段重要性计算数据的交换优先级, 车载端在收集数据的同时, 根据交换阈值与其他车载端进行数据交换。其次, 根据这些数据在本地建立当前数据与历史数据的索引, 以响应当前速度查询。进而, 时空范围划分成若干等价类, 通过数据中心的辅助, 采用联邦学习技术为每个等价类训练LSTM模型, 以支持未来速度查询。本方法为智能交通、智慧城市建设提供了一种新的解决方案。

参考文献

- [1] PELANIS M, ŠALTENIS S, and JENSEN C S. Indexing the past, present, and anticipated future positions of moving objects[J]. *ACM Transactions on Database Systems*, 2006, 31(1): 255–298. doi: 10.1145/1132863.1132870.
- [2] PFOSER D, BRAKATSOULAS S, BROSCHE P, et al. Dynamic travel time provision for road networks[C]. The 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Irvine, USA, 2008: 68. doi: 10.1145/1463434.1463513.
- [3] CHANG Jundong. Spatial-temporal based traffic speed

- imputation for GPS probe vehicles[C]. The Fifth International Conference on Network, Communication and Computing, Kyoto, Japan, 2016: 326–330. doi: [10.1145/3033288.3033339](https://doi.org/10.1145/3033288.3033339).
- [4] CHEN Rongsheng and LEVIN M W. Traffic state estimation based on Kalman filter technique using connected vehicle V2V basic safety messages[C]. 2019 IEEE Intelligent Transportation Systems Conference, Auckland, New Zealand, 2019: 4380–4385. doi: [10.1109/ITSC.2019.8917343](https://doi.org/10.1109/ITSC.2019.8917343).
- [5] 崔艳玲, 金蓓弘, 张扶桑. 基于数据融合的高速公路交通状况检测[J]. 计算机学报, 2017, 40(8): 1798–1812. doi: [10.11897/SP.J.1016.2017.01798](https://doi.org/10.11897/SP.J.1016.2017.01798).
- CUI Yanling, JIN Beihong, and ZHANG Fusang. Highway traffic condition detection with data fusion[J]. *Chinese Journal of Computers*, 2017, 40(8): 1798–1812. doi: [10.11897/SP.J.1016.2017.01798](https://doi.org/10.11897/SP.J.1016.2017.01798).
- [6] BEI Pan, DEMIRYUREK U, and SHAHABI C. Utilizing real-world transportation data for accurate traffic prediction[C]. IEEE 12th International Conference on Data Mining, Brussels, Belgium, 2012: 595–604. doi: [10.1109/ICDM.2012.52](https://doi.org/10.1109/ICDM.2012.52).
- [7] HE Zhixiang, CHOW C Y, and ZHANG Jiadong. STCNN: A spatio-temporal convolutional neural network for long-term traffic prediction[C]. The 20th IEEE International Conference on Mobile Data Management, Hong Kong, China, 2019: 226–233. doi: [10.1109/MDM.2019.00-53](https://doi.org/10.1109/MDM.2019.00-53).
- [8] TANG Keshuang, CHEN Siqu, CAO Yumin, *et al.* Short-term travel speed prediction for urban expressways: Hybrid convolutional neural network models[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(3): 1829–1840. doi: [10.1109/TITS.2020.3027628](https://doi.org/10.1109/TITS.2020.3027628).
- [9] TONG Guan, PENG Jiaheng, and LIANG Jun. Spatial-temporal graph multi-gate mixture-of-expert model for traffic prediction[C]. IEEE 26th International Conference on Intelligent Transportation Systems, Bilbao, Spain, 2023: 36–41. doi: [10.1109/ITSC57777.2023.10422031](https://doi.org/10.1109/ITSC57777.2023.10422031).
- [10] LI Maosen, CHEN Siheng, SHEN Yanning, *et al.* Online multi-agent forecasting with interpretable collaborative graph neural networks[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2024, 35(4): 4768–4782. doi: [10.1109/tnnls.2022.3152251](https://doi.org/10.1109/tnnls.2022.3152251).
- [11] 岳猛, 王怀远, 吴志军, 等. 云计算中DDoS攻防技术研究综述[J]. 计算机学报, 2020, 43(12): 2315–2336. doi: [10.11897/SP.J.1016.2020.02315](https://doi.org/10.11897/SP.J.1016.2020.02315).
- YUE Meng, WANG Huaiyuan, WU Zhijun, *et al.* A survey of DDoS attack and defense technologies in cloud computing[J]. *Chinese Journal of Computers*, 2020, 43(12): 2315–2336. doi: [10.11897/SP.J.1016.2020.02315](https://doi.org/10.11897/SP.J.1016.2020.02315).
- [12] QI Bozhao, KANG Lei, and BANERJEE S. A vehicle-based edge computing platform for transit and human mobility analytics[C]. The Second ACM/IEEE Symposium on Edge Computing, San Jose, USA, 2017: 1. doi: [10.1145/3132211.3134446](https://doi.org/10.1145/3132211.3134446).
- [13] 吴洪越, 陈志伟, 石博文, 等. 一种面向边缘计算环境的去中心化服务请求分发方法[J]. 计算机学报, 2023, 46(5): 987–1002. doi: [10.11897/SP.J.1016.2023.00987](https://doi.org/10.11897/SP.J.1016.2023.00987).
- WU Hongyue, CHEN Zhiwei, SHI Bowen, *et al.* Decentralized service request dispatching for edge computing systems[J]. *Chinese Journal of Computers*, 2023, 46(5): 987–1002. doi: [10.11897/SP.J.1016.2023.00987](https://doi.org/10.11897/SP.J.1016.2023.00987).
- [14] 张晓东, 张朝昆, 赵继军. 边缘智能研究进展[J]. 计算机研究与发展, 2023, 60(12): 2749–2769. doi: [10.7544/issn1000-1239.202220192](https://doi.org/10.7544/issn1000-1239.202220192).
- ZHANG Xiaodong, ZHANG Chaokun, and ZHAO Jijun. State-of-the-art survey on edge intelligence[J]. *Journal of Computer Research and Development*, 2023, 60(12): 2749–2769. doi: [10.7544/issn1000-1239.202220192](https://doi.org/10.7544/issn1000-1239.202220192).
- [15] WANG Tian, LUO Hao, ZHENG Xi, *et al.* Crowdsourcing mechanism for trust evaluation in CPCS based on intelligent mobile edge computing[J]. *ACM Transactions on Intelligent Systems and Technology*, 2019, 10(6): 62. doi: [10.1145/3324926](https://doi.org/10.1145/3324926).
- [16] FELDMAN M, LAI K, STOICA I, *et al.* May. Robust incentive techniques for peer-to-peer networks[C]. The 5th ACM Conference on Electronic Commerce, New York, USA, 2004: 102–111. doi: [10.1145/988772.988788](https://doi.org/10.1145/988772.988788).
- [17] 陈山枝, 葛雨明, 时岩. 蜂窝车联网(C-V2X)技术发展、应用及展望[J]. 电信科学, 2022, 38(1): 1–12. doi: [10.11959/j.issn.1000-0801.2022007](https://doi.org/10.11959/j.issn.1000-0801.2022007).
- CHEN Shanzhi, GE Yuming, and SHI Yan. Technology development, application and prospect of cellular vehicle-to-everything (C-V2X)[J]. *Telecommunications Science*, 2022, 38(1): 1–12. doi: [10.11959/j.issn.1000-0801.2022007](https://doi.org/10.11959/j.issn.1000-0801.2022007).
- [18] HOCHREITER S and SCHMIDHUBER J. Long short-term memory[J]. *Neural Computation*, 1997, 9(8): 1735–1780. doi: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [19] RAMSAK F, MARKL V, FENK R, *et al.* Integrating the UB-tree into a database system kernel[C]. The 26th International Conference on Very Large Data Bases, Cairo, Egypt, 2000: 263–272.
- [20] NIELSEN F. On a generalization of the Jensen–Shannon divergence and the Jensen–Shannon centroid[J]. *Entropy*, 2020, 22(2): 221. doi: [10.3390/e22020221](https://doi.org/10.3390/e22020221).
- [21] MCMAHAN B, MOORE E, RAMAGE D, *et al.* Communication-efficient learning of deep networks from decentralized data[C]. The 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, USA, 2017: 1273–1282.
- 韩京宇: 男, 教授, 研究方向为空间时态数据库、大数据与机器学习。
王彦之: 男, 硕士生, 研究方向为机器学习与数据挖掘。
陈进: 男, 硕士生, 研究方向为数据库系统与数据挖掘。
晏鑫鑫: 男, 硕士生, 研究方向为时空数据管理与数据挖掘。
张怡婷: 女, 副教授, 研究方向为大数据管理与网络安全。