文章编号: 1000-5641(2017)05-0186-15

基于实时路况的top-k载客热门区域推荐

吴 涛[1], 毛嘉莉[1], 谢青成[1], 杨艳秋[2], 王 锦[1]

(1. 西华师范大学 计算机学院, 四川 南充 637000;
2. 中国人民武装警察部队警官学院 电子技术系, 成都 610000)

摘要: 为降低城市出租车的空载率, 缓解路网交通拥堵压力, 亟需设计有效的出租车载客热门区域推荐方法. 针对传统的出租车相关推荐方法忽略实际路况导致推荐精度较低的现状, 提出了一个两阶段的载客热门区域实时推荐算法. 首先, 离线挖掘阶段, 基于出租车历史轨迹数据集提取基于时段属性的载客热门区域; 随后, 在线推荐阶段, 根据出租车请求位置及时间, 结合实时路况设计潜在空载时间开销函数 T_{cost} 对载客热门区域进行评测排序, 继而发现Top-k载客热门区域. 基于出租车轨迹数据集的实验结果表明, 结合实时交通状况的Top-k载客热门区域推荐方法以确保较小潜在空载时间开销, 相较于传统的出租车推荐方法具有较好的有效性与鲁棒性.

关键词: 潜在空载时间开销函数; 实际路况; 热门区域; 推荐
中图分类号: TP311 **文献标志码**: A **DOI:** 10.3969/j.issn.1000-5641.2017.05.017

Top-k hotspots recommendation algorithm based on real-time traffic

WU Tao[1], MAO Jia-li[1], XIE Qing-cheng[1], YANG Yan-qiu[2], WANG Jin[1]

(1. *College of Computer, China West Normal University, Nanchong Sichuan* 637000, *China*;
2. *Department of Electronic Technology, Officers College of PAP, Chengdu* 610000, *China*)

Abstract: To cut down the no-load rate of taxis and relieve the traffic pressure, an effective hotspot recommendation method of picking up passenger is necessitated. Aiming at the problem of lower recommendation precision of traditional recommendation technique due to ignoring the actual road situation, we propose a two-phase real-time hotspot recommendation approach for picking up passenger. In the phase of offline mining, time-based hotspots are extracted by mining the history taxi trajectory dataset. In the phase of online recommendation, according to the position and time of taxi requests, a potential no-passenger time cost evaluation function that based on real-time road situation is presented to evaluate and rank hotspots, and obtain top-k hotspots of picking up passenger.

收稿日期: 2017-06-19
基金项目: 四川省教育厅重点基金项目 (17ZA0381, 13ZA0015); 西华师范大学国家培育项目 (16C005); 西华师范大学英才科研基金 (17YC158)
第一作者: 吴 涛, 男, 硕士研究生, 研究方向为基于位置的服务. E-mail: 850517937@qq.com.
通信作者: 毛嘉莉, 女, 副教授, 硕士生导师, 研究方向为基于位置的服务. E-mail: maojl1231@163.com.

Experimental results on taxi trajectory data show that, our proposal ensure smaller potential no-load time overhead due to considering real-time traffic conditions, and hence has good effectiveness and robustness as compared to the traditional recommendation approached.

Key words: potential no-passenger time cost function; real-time traffic; hotspot; recommendation

0 引　　言

随着城市规模的不断扩大, 城市居民的出行需求日益增长, 出租车的数量急剧增多. 近年来随着网约车平台 (滴滴、优步) 的出现, 出租车行业在日趋激烈的市场竞争中面临着空载率高的严峻现实, 城市路网中大部分时段存在大量空载出租车为寻找乘客而漫无目的地在周边道路巡游; 与此同时, 城市的部分区域在特定时段因乘客爆发式激增使得出租车供不应求, 乘客面临着打车困难的问题. 这种方式不仅导致了大量的资源浪费, 同时也带来了不少负面影响, 如交通拥堵、环境污染等. 因此, 提高出租车的载客率, 保证其在较小开销下的盈利模式已成为智能交通管理的一个有力措施. 随着智能交通系统的发展, 感知设备如车载 GPS (Global Positioning System) 在出租车中得到了普及, 这些设备以一定的时间间隔持续不断地向出租车信息管理中心发送出租车的位置信息, 包括时间戳、经纬度坐标、载客状态、速度等数据. 这些数以万计的数据中蕴含着丰富的乘客及出租车司机的行为特征信息, 通过对这些数据进行挖掘分析可以发现出租车运行轨迹的潜在时空特性, 如发现载客热门区域[1-2]、载客热点[3-4]等 (如图 1 中的红色标注点 Hotspot 1、Hotspot 2、Hotspot 3), 当出租车司机于某地点 (如图 1 中的蓝色标注点 Anxin mansion) 提出载客推荐请求时, 可向其推荐距离较近的城市载客热门区域, 以提高出租车司机的客源寻找效率, 解决出租车载客率低的问题.

在传统的出租车相关推荐系统中[1-5], 多数研究忽略了实时交通对推荐效果的影响, 在推荐过程中采用基于距离最近或基于载客概率最大的原则对出租车司机进行推荐, 如图 1 中, 载客热门区域 Hotspot 1 因与出租车当前位置 Anxin mansion 的路段距离相较于 Hotspot 2、Hotspot 3 稍大, 或由于其载客概率小于 Hotspot 2、Hotspot 3, 在推荐过程中时常被认为劣于 Hotspot 2、Hotspot 3, 甚至被舍弃. 然而, 结合实际交通情况分析后发现, 基于位置 Anxin mansion 到 Hotspots 2、Hotspot 3 的部分路段在当前时段拥塞状况严重 (图中标记为“红色”的路段为拥堵路段), 若将 Hotspot 2、Hotspot 3 对出租车司机进行推荐, 不仅将导致出租车空载行驶时间过长, 降低工作效率, 同时进一步加重了城市交通负担, 使得城市交通形势更加严峻.

因此, 针对以上传统出租车相关推荐系统所面临的问题, 本文在载客热门区域推荐时考虑结合实时的交通路况信息, 构建并实施了一个两阶段的实时载客热门区域推荐算法: ① 离线挖掘阶段, 通过对出租车历史轨迹数据的深度挖掘提取基于不同时段, 区分工作日、休息日的载客热门区域; ② 在线推荐阶段, 根据数据中心实时到达的出租车轨迹数据流分析最新时段的道路交通情况, 结合实时交通状况进行 top-k 载客热门区域推荐, 从根本上解决出租车载客率低、乘客打车难、出租车空载时间长等难题, 缓解道路拥堵, 进一步推进智能交通建设, 本文的主要贡献如下.

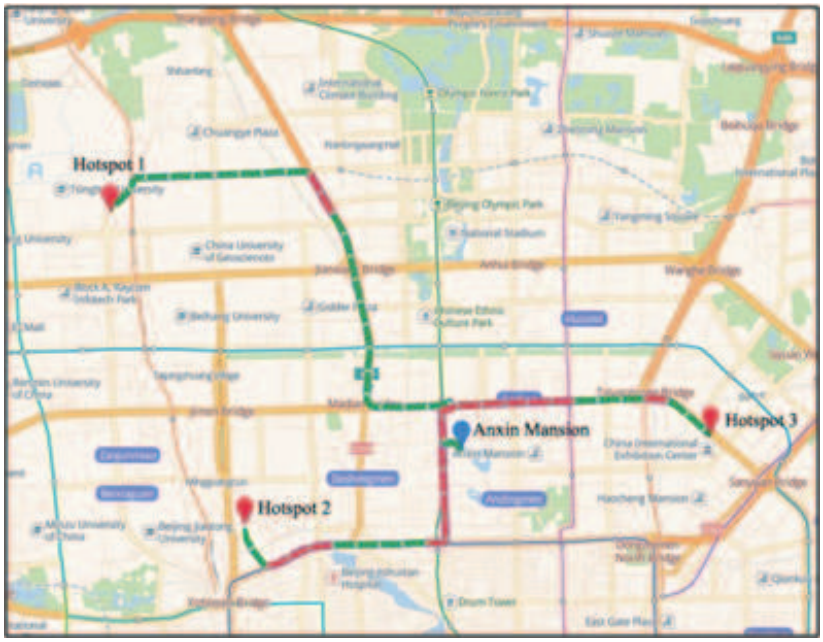


图 1 载客热门区域推荐示例

Fig. 1 The examples of hotspots recommendation

(1) 本文提出了一个两阶段的出租车载客热门区域实时推荐算法, 包括离线挖掘和在线推荐两个阶段. 首先, 在离线挖掘阶段, 根据出租车历史轨迹数据发现载客热门区域集合; 其次, 在在线推荐阶段, 根据当前时间戳结合载客热门区域的时段属性遴选出候选载客热门区域集; 最后, 设计潜在空载时间开销函数 T_{cost} 对各载客热门区域进行评测, 完成 top-k 推荐.

(2) 针对当下各城市的交通拥堵情况形势严峻, 而现有的出租车相关推荐研究忽视了城市路网的实时交通状况使得推荐精度较低. 本文通过设计潜在空载时间开销函数 T_{cost}, 将出租车载客热门区域推荐问题转化为对载客热门区域潜在空载时间开销的计算, 在推荐过程中考虑了实时的交通情况. 基于实际出租车轨迹数据集的实验验证了本文提出的载客热门区域推荐算法的准确性和有效性.

1 相关工作

目前基于轨迹数据挖掘的出租车相关推荐研究已成为国内外的研究热点之一. 2014 年, Qu[3] 等人基于最大收益化原则, 为出租车司机推荐了由一系列载客热点排列形成的寻客路线, 将推荐问题转化为移动序列推荐 (Mobile Sequence Recommendation, MSR) 问题; 文献 [4] 从出租车轨迹中提取载客热点, 并利用出租车历史轨迹数据集构建概率模型, 为出租车司机推荐到达载客热点的路径以最大概率搭载到下一位乘客. 以上研究忽视了不同时段对载客热点的影响. 考虑到出车司机的不同寻客策略, 文献 [5] 基于大规模出租车轨迹数据集, 采用支持向量机模型发现了高效的寻客方案, 并结合时空特性对缺乏经验的出租车司机进行推荐; 文献 [2] 主要基于 4 000 多条历史轨迹数据集发现载客热门区域, 并通过改进的 ARIMA 方法预测乘客分布特征完成对出租车司机的推荐; 文献 [1] 根据时间、天气、出租车地理位置的上下文信息对出租车的需求分布进行预测, 设计了热度值函数作为各载客热门区域的评价标准. 同时, 也有研究[6] 基于出租车轨迹流数据, 预测在短时间范围内的出

租车乘客的空间分布特征. 当然, 也有部分研究致力于为乘客的服务, 如文献 [7] 中基于不同时段、是否为工作日以及天气情况等3个因素, 采用基于朴素贝叶斯分类器的方法预测空载出租车数量. 齐观德等[8]使用出租车轨迹历史数据, 预测乘客在某时某地等候出租车需要的时间. 以上出租车相关推荐服务的研究中, 在对出租车司机进行推荐的过程中主要基于载客概率最大或驾驶距离最短等推荐原则, 却忽略了实际交通情况对推荐效果的影响.

针对以上传统出租车相关推荐研究的不足, 本文结合路网实际交通状况对出租车司机进行载客热门区域推荐服务, 实时交通路况的评测也就作为另一个与本文相关的工作. 由于出租车随机性、动态性的特点, 其运行状态能很好地反映城市的交通情况, 因此Mao等[9]通过对实时到达的出租车轨迹流进行聚类分析, 在无路网数据的情况下判断交通拥塞路段. 文献 [10] 中, 通过对出租车轨迹数据的挖掘, 分析了北京市的交通拥堵情况. Han[11]等结合路网数据, 设计了NEAT-a-road-network识别方法, 快速准确地将移动对象的轨迹进行聚类, 以识别道路拥堵路段情况. 以上研究主要基于出租车轨迹数据并应用于城市路网中交通情况的预测及判断, 不仅有利于政府部门对城市路网的规划, 还能为乘客及司机的出行提供建议.

本文结合实时路况对出租车司机进行载客热门区域的推荐, 不仅能提高出租车的载客率, 同时也有利于城市交通, 进一步推进智能城市的建设.

2　问题描述

本文将城市路网表示为一个图 (后文简称"路网图"), $G = \langle V, E\rangle$. 提取路段交叉口或路段终点作为路网图中的结点集合V, 而路网图中的有向边集E则为实际道路的映射, 并将路径定义如下.

定义 1 (路径)　路径R为一组起始位置P_{start}和终止位置P_{end}之间的有序路段交点序列, $R = P_{\text{start}} \to P_1 \to P_2 \to P_3 \to \cdots \to P_n \to P_{\text{end}}$, 并且满足边$e_{P_i,P_{i+1}} \in G.E$, $1 \leqslant i \leqslant n$.

定义 2 (出租车轨迹)　出租车轨迹Tr是指该出租车基于时间的有序GPS采样点序列, $\text{Tr} = p_1, \cdots, p_i, \cdots, p_n$. 其中每个GPS采样点$p_i (1 \leqslant i \leqslant n)$包括位置数据$p_i$.lon、$p_i$.lat以及出租车位于该位置的时间戳$p_i.t$, 载客状态$p_i.s$, 速度$p_i.v$, 角度$p_i.b$等信息, 且满足$\forall i < j$, $p_i.t < p_j.t$.

定义 3 (载客热门区域)　载客热门区域H是指某个连续时段内出租车发生载客事件较为密集的城市区域, 该区域为任意形状, 并且具有时段属性$T_{\text{duration}} = \langle T_{\text{start}}, T_{\text{end}}\rangle$.

不同于传统的载客热门区域挖掘方法, 本文考虑不同时段内载客点分布的规律性, 为载客热门区域定义时段属性: 在其对应的时段T_{duration}内, 该区域被视为载客热门区域, 而超过时段T_{duration}时, 则不再被视为载客热门区域. 因此, 在对出租车司机进行推荐时, 应结合当前时间戳遴选出候选载客热门区域, 如定义4.

定义 4 (候选载客热门区域)　候选载客热门区域为载客热门区域的延伸定义, 给定时间戳T, 如果某载客热门区域的时段属性T_{duration}满足$T_{\text{start}} \leqslant T < T_{\text{end}}$, 则该载客热门区域为基于时间$T$的候选载客热门区域.

出租车司机于不同的载客热门区域主要有两种寻客方式: 第一, 出租车司机在载客热门区域中以驻车或排队的方式被动性地等待乘客前来搭载, 这种方式通常发生在乘客客流量呈现较强规律性变化或乘客以集中式出现的载客热门区域, 如火车站、飞机场等; 第二, 出租车司机在载客热门区域中以巡游的方式自主性地寻找乘客. 这种情况通常发生在地理范围较大、乘客分布较分散的载客热门区域, 如某大型景点, 公园等. 基于以上两种不同的寻

客方式, 本文将出租车以空载状态自进入载客热门区域起, 至在该区域搭载到下一位乘客止所经历的时间定义为候客时间, 如定义5.

定义 5 (候客时间) 候客时间 T_{wait} 是指出租车在 t_0 时刻由空载状态进入某载客热门区域 H 起直到 t_1 时刻在该区域搭载到下一位乘客所经历的时间. 即 $T_{\text{wait}} = t_1 - t_0$, 且满足 $0 \leqslant T_{\text{wait}} \leqslant t_{\max}$, 其中 $t_{\max}$ 为人为定义的最长候客时间 (本文取 $t_{\max} = 30$ min). 具体公式为

$$T_{\text{wait}} = \begin{cases} t_1 - t_0, & t_1 - t_0 \in [0, t_{\max}], \\ \text{undefined}, & t_1 - t_0 \in (-\infty, 0) \cup (t_{\max}, +\infty). \end{cases} \tag{1}$$

式 (1) 中若出租车由空载状态进入该区域经历 $t_{\max}$ 后还处于空载状态, 或者在该区域经历时间不到 $t_{\max}$ 便由空载状态离开该区域, 则不对以上情况的候客时间进行定义.

定义 6 (空载时间开销) 空载时间开销 T_{cost} 指出租车由载客状态为空载状态起直到下一次跳变为载客状态止所经历的时间.

对于多数载客热门区域而言, 其空载时间开销值 T_{cost} 主要依赖于实时的路段交通情况, 若出租车当前位置到载客热门区域的路段交通拥塞情况较严重, 则其相应的空载时间开销值 T_{cost} 越大. 因此, 本文将对出租车司机进行载客热门区域的推荐转化为对各载客热门区域的空载时间开销值 T_{cost} 的评测问题. 同时考虑部分载客热门区域需要以排队方式搭载乘客, 或在到达该区域后不能及时地搭载到下一位乘客, 如火车站, 飞机场等. 因此本文将空载时间开销 T_{cost} 视为由出租车当前位置到载客热门区域的时间和候客时间组成.

定义 7 (问题定义) 基于实时路况的 top-k 载客热门区域推荐是指给定出租车当前位置 P_{current}, 提交推荐请求的时间戳 T_{current}, 基于时间戳 T_{current} 的一系列候选载客热门区域集 $H = \{h_1, h_2, \cdots, h_i, \cdots, h_n\}$, 推荐算法为出租车司机推荐 k 个潜在空载时间开销 T^i_{cost} 值最小的载客热门区域, 即

$$h = \text{argmin}_{\text{Top-}k:h_i \in H} T^i_{\text{cost}}(h_i, P_{\text{current}}, T_{\text{current}}), \quad 1 \leqslant i \leqslant n. \tag{2}$$

不同于传统的出租车相关推荐研究, 本文通过对载客热门区域潜在空载时间开销值的评测, 结合实时路况对出租车司机进行推荐, 使得推荐的载客热门区域与出租车当前位置的交通状况良好. 基于以上对推荐问题的定义可知, 发现载客热门区域以及如何计算载客热门区域的潜在空载时间开销 T^i_{cost} 成为本文考虑的首要问题. 后面将对这两部分的内容进行详细阐述.

3 推荐模型

3.1 算法框架

本文构建了一个两阶段的载客热门区域推荐算法, 框架如图2所示. 首先, 在离线挖掘部分, 结合出租车历史轨迹数据集, 采用改进的 DBSCAN 聚类方法发现区别工作日、休息日且具有时段属性的载客热门区域, 这些载客热门区域以凸多边形的方式进行存储, 并且随着出租车轨迹流的不断积累每隔一定时间 (如 1 month) 进行更新维护; 其次, 第二阶段为在线推荐部分, 当出租车司机于某地点完成一次载客交易后提出推荐请求, 系统将自动获取出租车的位置信息以及当前时间戳, 根据当前时间戳遴选出候选载客热门区域, 同时结合实时到达的出租车轨迹流完成对各路段的实际拥塞情况的评测, 进一步计算出租车当前位置到各候选载客热门区域的空载穿行时间, 然后基于出租车历史轨迹数据集预测各候选载客热门

区域的候客时间, 最后结合空载穿行时间和候客时间完成对各候选载客热门区域的潜在空载时间开销 T_{cost} 的评测完成top-k推荐.

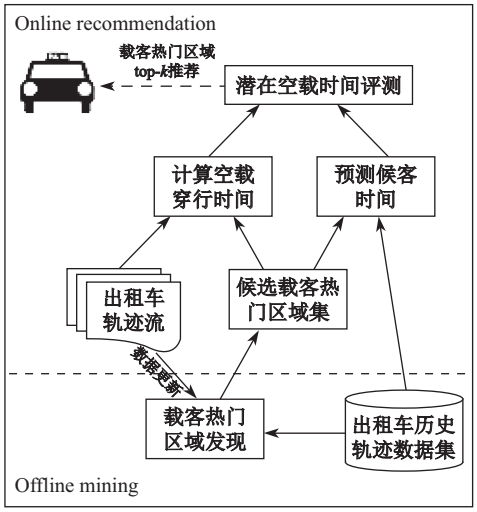


图 2　出租车载客热门区域推荐模型框架图

Fig. 2　Framework of hotspot recommendation

3.2　基于历史轨迹的载客热门区域提取

考虑区别工作日、休息日, 一天中的不同时段对载客热门区域的影响, 本文将出租车历史轨迹数据划分为工作日以及休息日两个部分, 并将一天分为12个时间片 {00:00-01:00, 01:00-02:00, ⋯, 22:00-23:00, 23:00-24:00}, 采用改进的DBSCAN聚类方法进行分时段的聚类完成对载客热门区域的提取. 如图3所示.

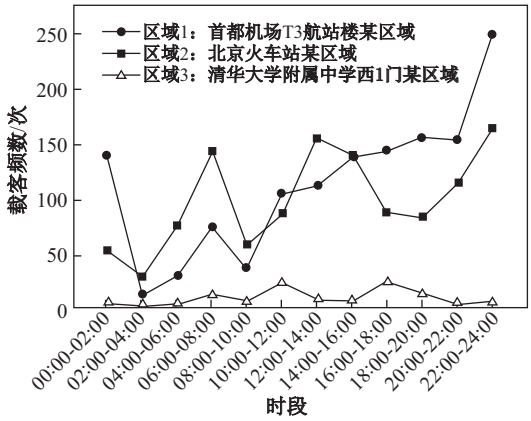


图 3　北京市机场、火车站、中学区域各时段平均载客频数

Fig. 3　The average passenger frequency of airport, train station and school

载客热门区域作为本文的推荐元素, 如定义3中为出租车在某个连续时段内发生载客事件较为密集的区域, 即在该区域中发生载客事件概率较大, 因此将该区域作为候客地点向出租车司机进行推荐. 这些区域通常具有特定的人群移动规律和功能特征, 如可能代表一个商圈, 或者是一个学校、景点等, 且通常具有时段属性. 结合定义3旨在发现某一连续时段内载客点密度较大的区域, 如对于经验丰富的出租车司机而言, 通常能清楚火车

站各班次火车到站时间、电影院电影放映结束的时段或者学校的放学时间, 由于在这些时段内该区域发生载客事件较频繁, 因此出租车司机通常选择对应时段属性的载客热门区域作为候客地点. 如图3, 由于深夜通往市区的公交、地铁停运, 位于郊区的机场在晚上23:00点至凌晨00:00点被视为载客热门区域, 即机场作为载客热门区域的时段属性可被定义为$T_{\text{duration}} = \langle 23:00, 24:00\rangle$.

本文通过从历史出租车轨迹数据集中提取出载客状态由空载状态跳变为载客状态的GPS采样点, 将此点作为出租车载客点. 基于上文中对载客热门区域的定义和分析, 本文采用改进的DBSCAN聚类算法对提取出的载客点进行时空聚类. 该方法将传统的DBSCAN聚类算法在时空域上进行扩展, 将载客点分布于时空立体空间中进行聚类, 通过两个参数$\varepsilon_{\text{temporal}}$和$\varepsilon_{\text{spatio}}$分别作为时间维度和空间维度的度量半径, 如图4所示, 不仅考虑了空间属性的相似性, 同时也考虑了时间维度的相似性. 采用时空密度作为实体空间相似性的度量标准, 将时空簇视为一系列由低密度区域(噪声)分割开的高密度连通区域. 由于改进的DBSCAN聚类算法在聚类过程中考虑了时间维度上的相似性, 使得聚类结果为在某一连续时段内数据点密度较大的的时空簇. 即位于同一个簇中的载客点不仅在地理位置上邻近, 在时间维度上也是较邻近的, 最终达到在某一连续时间段和密度较大的聚类目标. 伪代码如算法1 (Algorithm 1) 所示.

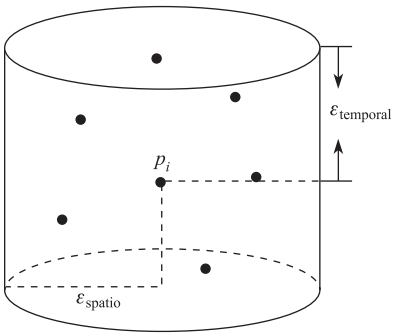


图4 时空邻近域

Fig. 4 Space-time adjacency domain

算法1的基本时间复杂度为$O(n^2)$, 若对于数据结构如KD树, 由于其可以有效检索特定点给定距离内的所有点, 其时间复杂度便降低到$O(n\log n)$. 由于其中每个点只需要维持少量数据, 即簇标号和每个点的标识, 其空间复杂度为$O(n)$.

注意算法1中为评估各载客点在时空属性上的相似性, 本文采用地球球面距离作为其在空间属性上的相似度度量函数, 两个载客点p_i和p_j的地球球面距离计算公式为

$$D_{\text{sptio}}(p_i, p_j) = R \cdot \arccos\Big(\sin(p_i.\text{lat})\sin(p_j.\text{lat}) + \cos(p_i.\text{lat})\cos(p_j.\text{lat})\cos(p_i.\text{lon} - p_j.\text{lon})\Big), \quad (3)$$

其中, R为地球的近似半径, 取$R = 6\,370.996\,81$ km.

将载客点在时间属性上的相似性度量函数定义为

$$D_{\text{temporal}}(p_i, p_j) = \begin{cases} |p_i.t - p_j.t|, & |p_i.t - p_j.t| \leqslant 12\text{ h}, \\ 24 - |p_i.t - p_j.t|, & |p_i.t - p_j.t| > 12\text{ h}. \end{cases} \quad (4)$$

Algorithm 1 Hotspot generation algorithm based on improved DBSCAN

Input: A list of pick-up locations point pointsList=$\{p_1, p_2, \cdots, p_n\}$,
MinPts,$\varepsilon_{\text{temporal}}, \varepsilon_{\text{spatio}}$
Output: Clustering result clusterList
1: Initialize pointsList(clusterID←0, isVisited←false, isNoised←false);
2: cluster_i ← 1;
3: for each point p_i in pointsList do
4: if(!p_i.isVisited)then
5: p_i.isVisited=true;
6: find the Neiborhood points for p_i about $\varepsilon_{\text{temporal}}$, $\varepsilon_{\text{spatio}}$: $N_{\varepsilon_{\text{temporal}},\varepsilon_{\text{spatio}}}p_i$;
7: if($|N_{\varepsilon_{\text{temporal}},\varepsilon_{\text{spatio}}}p_i| < \text{MinPts}$)then
8: p_i.isNoised=true;
9: else
10: p_i.clusterID=cluster_i;
11: for each point $p'_{i.\text{adjacent}}$ in $N_{\varepsilon_{\text{temporal}},\varepsilon_{\text{spatio}}}$ p_i do
12: if(!$p'_{i.\text{adjacent}}$.isVisited)then
13: $p'_{i.\text{adjacent}}$.isVisited=true;
14: find the Neiborhood points for $p'_{i.\text{adjacent}}$ about $\varepsilon_{\text{temporal}}, \varepsilon_{\text{spatio}}$: $N_{\varepsilon_{\text{temporal}},\varepsilon_{\text{spatio}}}p'_{i.\text{adjacent}}$;
15: if($|N_{\varepsilon_{\text{temporal}},\varepsilon_{\text{spatio}}}p'_{i.\text{adjacent}}| \geqslant \text{MinPts}$)then
16: $N_{\varepsilon_{\text{temporal}},\varepsilon_{\text{spatio}}}p_i = N_{\varepsilon_{\text{temporal}},\varepsilon_{\text{spatio}}}p_i \cup N_{\varepsilon_{\text{temporal}},\varepsilon_{\text{spatio}}}p'_{i.\text{adjacent}}$;
17: end if
18: end if
19: if($p'_{i.\text{adjacent}}$.clusterID=0)then
20: $p'_{i.\text{adjacent}}$.clusterID=cluster_i;
21: if($p'_{i.\text{adjacent}}$.isNoised)then
22: $p'_{i.\text{adjacent}}$.isNoised=false;
23: end if
24: end if
25: end for
26: cluster_i++;
27: end if
28: end if
29: end for

基于以上改进的DBSCAN聚类算法, 区别工作日、休息日, 将提取出的载客点按其时间戳分为24个时间片进行时空聚类, 生成各时间片中时空密度较高的出租车载客点集合, 其聚类结果即时空簇便为某个连续时段内载客事件发生较为频繁的区域. 本文采用Graham扫描法(Graham Scan)[12], 发现各时空簇在空间属性上的凸包 (Convex Hull), 以凸多边形表示载客热门区域的几何属性, 实现对时空簇到地理几何数据的转化, 同时计算出各时空簇的簇心, 代表该载客热门区域对出租车司机进行推荐, 最后将该时空簇所处于的时间片作为其时段属性.

随着城市的扩展以及变迁, 载客热门区域的分布也随之变化. 因此随着出租车轨迹数据的积累, 载客热门区域应每隔一定时间 (如1 month) 进行增量式更新维护, 以更准确地获取发生变化或新增加的载客热门区域.

3.3 在线推荐

3.3.1 潜在空载时间开销函数

基于以上载客热门区域提取办法, 准确获取到区分工作日、休息日中各时段的载客热门区域集, 该部分工作以离线的方式进行处理. 在线推荐阶段中, 根据当前时间戳T_{current}遴选基于T_{current}的候选载客热门区域, 然后对各候选载客热门区域的潜在空载时间开销T_{cost}进行计算. 本节内容主要介绍潜在空载时间开销函数T_{cost}的构造及计算方法.

实际生活中, 出租车司机主要以快速搭载到下一位乘客以此提高工作效率作为其选择候客

地点的主要因素. 考虑如下情况, 出租车司机在完成一次载客交易后如何能以最短时间搭载到下一位乘客, 通常会考虑以下两个因素.

(1) 面对日益庞大的城市交通网络, 道路拥堵的情况时有发生, 尤其在大城市会更为普遍. 因此出租车司机会选择距离当前位置较近且交通状况良好的载客热门区域, 从而避开交通拥堵路段减少出租车的空载穿行时间.

(2) 考虑不同功能特征的载客热门区域, 其候客时间也存在较大区别, 如火车站人流量大且相对集中, 其候客时间稍长; 而对于公园、商场等由于其乘客分布较为分散, 候客时间则较短. 当然对于空载出租车司机而言, 通常会选择候客时间较短的载客热门区域作为其候客的区域, 以最短时间搭载到下一位乘客, 提高工作效率.

本文结合以上两个因素, 对于一个基于位置 P_{current}、当前时间戳为 T_{current} 的出租车司机而言, 能知道处于当前时间戳 T_{current} 的实际路段交通状况以及各载客热门区域的候客时间 T_{wait}, 基于最短时间搭载到下一位乘客作为其选择载客热门区域的策略. 本文设计潜在空载时间开销函数 T_{cost} 对基于时间戳 T_{current} 的候选载客热门区域进行评测, 公式为

$$T_{\text{cost}} = T_{\text{drive}} + T_{\text{wait}}. \tag{5}$$

式 (5) 中 T_{drive} 表示空载穿行时间, 即当前位置 P_{current} 到候选载客热门区域的通行时间, 若基于当前位置到候选载客热门区域的路段拥塞情况严重, 相应地在该路段的穿行时间则越长. 由于道路的复杂性, 交通事故时有发生, 路段的拥塞情况是无规律可循的, 因此空载穿行时间是基于当前时间 T_{current} 的, 需要对路网中交通状况进行实时评测, 详见第 3.3.2 节. T_{wait} 表示候客时间, 由于载客热门区域的候客时间的预测与时段有关, 即候客时间是基于时间 $T_{\text{current}} + T_{\text{drive}}$ (即预计到达载客热门区域的时间) 所处时段的, 计算方法详见第 3.3.3 节. 因此 T_{drive} 和 T_{wait} 并不是相互独立的. 下文将对空载穿行时间和候客时间的预测方法进行详细阐述.

3.3.2 空载穿行时间预测

空载穿行时间 T_{drive} 作为潜在空载时间开销评测函数 T_{cost} 的组成部分之一, 可被理解为出租车当前位置到载客热门区域的时间, 其值主要由实时的路段交通情况决定, 同时, 由于多数出租车司机通常选择两位置间的最短路径行驶, 因此本文将候选载客热门区域 h_i 的簇心 h_i^{cen} 与当前位置 P_{current} 之间的最短路径作为出租车到 h_i 的候选路径, 并结合该路径所包含的路段交通情况完成空载穿行时间的预测.

结合路网图 $G = \langle V, E\rangle$, 基于起始位置 P'_{start} 和终止位置 P'_{end} 的最短路径 R_{short} 可通过如下步骤进行计算.

(1) 根据位置点 P'_{start}, P'_{end} 的经纬度坐标分别将其匹配到距离最近的路段 $e_{P_{m.s},P_{n.s}}$, $e_{P_{m.e},P_{n.e}}$, 且 $e_{P_{m.s},P_{n.s}}$, $e_{P_{m.e},P_{n.e}} \in G.E$.

(2) 将路网图 G 做以下临时操作: 删除边 $e_{P_{m.s},P_{n.s}}$, $e_{P_{m.e},P_{n.e}}$, 添加临时顶点 P_{start}, P_{end}, 临时边 $e_{P_{m.s},P_{\text{start}}}$, $e_{P_{\text{start}},P_{n.s}}$, $e_{P_{m..e},P_{\text{end}}}$, $e_{P_{\text{end}},P_{n.e}}$.

(3) 基于临时更改后的路网图, 采用 Dijkstra 算法[13]作为寻路策略, 完成对基于起始位置 P'_{start} 到终止位置 P'_{end} 之间的最短路径 R_{short} 的计算.

基于以上最短路径的计算方法, 我们将获取基于出租车当前位置 P_{current} 到各候选载客热门区域的最短路径 $\text{Path}_i = P_{\text{current}} \to P_1 \to P_2 \to \cdots \to P_n \to h_i^{\text{cen}}$, 最后根据该路径中各路段的实际交通状况计算平均速度, 完成对空载穿行时间 T_{drive} 的计算, 公式为

$$T_{\text{drive}} = \frac{D(P_{\text{current}}, P_1)}{\text{speed}_{P_{\text{current}},P_1}} + \frac{D(P_n, h_i^{\text{cen}})}{\text{speed}_{P_n,h_i^{\text{cen}}}} + \sum_{i=1}^{n-1} \frac{D(P_i, P_{i+1})}{\text{speed}_{P_i,P_{i+1}}}, \tag{6}$$

其中, $D(P_{\text{current}}, P_1)$表示当前位置$P_{\text{current}}$到道路交叉口$P_1$的距离, $\text{speed}_{P_{\text{current}},P_1}$为当前位置$P_{\text{current}}$所处路段基于当前时间的平均速度, 同理, $D(P_n, h_i^{\text{cen}})$、$\text{speed}_{P_n,h_i^{\text{cen}}}$分别表示道路交叉口$P_n$到候选载客热门区域$h_i$的距离、候选载客热门区域$h_i$位于路段的平均速度, $D(P_i, P_{i+1})$表示路段$e_{P_i \to P_{i+1}}$的长度, $\text{speed}_{P_i,P_{i+1}}$表示路段$e_{P_i \to P_{i+1}}$的基于时间$T_{\text{current}}$的平均速度.

本文通过基于滑动窗口的轨迹流聚类方法[9]获得各路段基于时间T_{current}的实时平均速度, 该方法通过将实时到达的出租车轨迹数据流进行增量式聚类, 并且能删除过时数据 (即离时间戳T_{current}较远的轨迹数据) 对聚类结果的影响, 因此该聚类结果可作为当前时段各路段交通状况的评测指标.

3.3.3　候客时间预测

候客时间作为潜在空载时间开销评测函数T_{cost}的另一个组成部分, 本文采用基于朴素贝叶斯分类器[14]的方法对载客热门区域的候客时间进行预测.

不同的载客热门区域在不同的时段候客时间不同, 同时还考虑到是否为工作日对载客热门区域候客时间也有影响, 即对于候选载客热门区域h_i, 由于候客时间受时间段T_{period}以及是否为工作日D的影响. 根据贝叶斯定理, 该候选载客热门区域h_i候客时间T_{wait}为y_i的后验概率为

$$P(T_{\text{wait}} = y_i \,|T_{\text{period}}, D\,) = \frac{P(T_{\text{wait}} = y_i)P(T_{\text{period}}, D\,|T_{\text{wait}} = y_i\,)}{P(T_{\text{period}}, D)}. \tag{7}$$

将1 d分为24个时间段, 则T_{period} ={00:00-01:00, 01:00-02:00, $\cdots$, 22:00-23:00, 23:00-24:00}, 并且区别工作日、休息日D = {workday, weekend}. 作为载客热门区域, 其候客时间当然不可能是无限大的, 即如果候客时间大于最长候客时间$t_{\max}$, 则认为推荐失败. 因此我们将候客时间$[0, t_{\max}]$平均分为m个时间桶 (本文设置$t_{\max} = 30\,\text{min}, m = 30$), 则时间桶$y_i$取值为

$$\begin{cases} \Delta t = t_{\max}/m, \\ y_i = ((i-1)\Delta t, i\Delta t), \quad i = 1, 2, 3, \cdots, m. \end{cases} \tag{8}$$

基于y_i的取值, 载客热门区域的候客时间T_{wait}的预测则转化为y_i的极大后验概率, 表示的公式为

$$T_{\text{wait}} = \underset{y_i, i=1,2,3,\cdots,m}{\arg\max} P(T_{\text{wait}} = y_i | T_{\text{period}}, D). \tag{9}$$

结合公式 (7) 中$P(T_{\text{period}}, D)$对于y_i的不同取值是固定的, 因此转化为

$$T_{\text{wait}} = \underset{y_i, i=1,2,3,\cdots,m}{\arg\max} P(T_{\text{wait}} = y_i)P(T_{\text{period}}, D\,|T_{\text{wait}} = y_i\,). \tag{10}$$

由于变量T_{period}、T_{wait}之间相互独立, 互不影响, 因此公式 (10) 最后做转化为

$$T_{\text{wait}} = \underset{y_i, i=1,2,3,\cdots,m}{\arg\max} P(T_{\text{wait}} = y_i)P(T_{\text{period}}\,|T_{\text{wait}} = y_i\,)P(D\,|T_{\text{wait}} = y_i\,). \tag{11}$$

根据公式 (11), 将历史出租车轨迹数据作为训练集, 给定候选载客热门区域h_i, 时间段T_{period}以及是否为工作日D, 我们将能对该候选载客热门区域的候客时间T_{wait}进行预测. 其中时间段T_{period}主要依赖于当前时间T_{current}以及潜在空载行驶时间T_{drive}, 即T_{period}为$T_{\text{current}} + T_{\text{drive}}$对应的时段获得. 并且公式 (11) 中先验概率$P(T_{\text{wait}} = y_i)$、条件概率$P(T_{\text{period}}\,|T_{\text{wait}} = y_i\,)$、$P(D\,|T_{\text{wait}} = y_i\,)$都可通过计算训练集中相应事件所占的比例进行估计, 其计算公式为公式 (12)、(13)、(14).

$T_{\text{wait}} = y_i$ 表示在该载客热门区域中搭载到下一位乘客并且候客时间 T_{wait} 在时间桶 y_i 中的事件, 则有

$$P(T_{\text{wait}} = y_i) = \frac{\Gamma_0^{24}(0 \sim 1; (t_1 - t_0) \in y_i)}{\Gamma_0^{24}(0 \sim)}. \tag{12}$$

对于一个载客热门区域 h_i, 由于其作为多边形的方式进行存储, 提取与该多边形有交点的出租车历史轨迹, 得到出租车到达该载客热门区域的时间 t_0, 以及载客状态跳变为"1"的时间 t_1. 因此, 出租车在该载客热门区域的候客时间 $T_{\text{wait}} = t_1 - t_0$ (如定义 5). 公式 (12) 中 $\Gamma_0^{24}(0 \sim 1; (t_1 - t_0) \in y_i)$ 表示在载客热门区域 h_i 中, 出租车搭载到下一位乘客并且候客时间 T_{wait} 在时间桶 y_i 中发生的数量; $\Gamma_0^{24}(0 \sim)$ 表示在载客热门区域 h_i 中, 出租车以空载状态进入该载客热门区域发生的数量. 同理, 根据贝叶斯公式, 条件概率可表示为

$$P(T_{\text{period}} \,|T_{\text{wait}} = y_i\,) = \frac{\Gamma_{T_{\text{period}}}(0 \sim 1; (t_1 - t_0) \in y_i)}{\Gamma_0^{24}(0 \sim 1; (t_1 - t_0) \in y_i)}, \tag{13}$$

$$P(D \,|T_{\text{wait}} = y_i\,) = \frac{\Gamma_0^{24}(0 \sim 1; (t_1 - t_0) \in y_i; D)}{\Gamma_0^{24}(0 \sim 1; (t_1 - t_0) \in y_i)}. \tag{14}$$

公式 (13) 中 $\Gamma_{T_{\text{period}}}(0 \sim 1; (t_1 - t_0) \in y_i)$ 表示在载客热门区域 h_i 中, 在时间段 T_{period} 中, 出租车搭载到下一位乘客并且候客时间T_{wait} 在时间桶 y_i 中的数量; 公式 (14) 中 $\Gamma_0^{24}(0 \sim 1; (t_1 - t_0) \in y_i; D)$ 表示在载客热门区域 h_i 中, 出租车搭载到下一位乘客并且候客时间 T_{wait} 在时间桶 y_i 中发生日期为 D 的数量.

3.3.4 top-k 推荐

基于以上对空载时间开销函数 T_{cost} 的构造及计算方法的描述, 本文将对基于当前时间 T_{current} 的候选载客热门区域进行空载时间开销的评测, 即候选载客热门区域的空载时间开销值越小, 则出租车当前位置到该候选载客热门区域的路段交通情况越好, 出租车司机于该载客热门区域将能以更短的时间搭载到下一位乘客. 解决了传统出租车相关推荐研究忽视了实际交通状况的缺陷. 同时, 为了避免"若处于相同位置的多辆出租车司机同时发出推荐请求时, 推荐系统的推荐结果为同一载客热门区域"这一情况, 本文将空载时间开销值最小的前 k 个载客热门区域对出租车司机进行推荐以供其挑选.

4 实 验

4.1 实验环境及数据集

本文实验采用北京某地区 2013 年 10 月共 30 d 内的出租车轨迹数据集, 该数据集包含近 30 000 多辆出租车的真实 GPS 轨迹数据. 本文作为一个两阶段的推荐系统, 其中离线挖掘部分基于历史数据发现载客热门区域, 在线推荐部分则需要通过出租车的实时轨迹流分析路网交通情况. 因此实验将数据集分为离线数据集 C1、在线数据集 C2. 利用离线数据集 C1 发现载客热门区域并作为载客热门区域候客时间的训练集对候客时间进行预测, 并利用在线数据集 C2 模拟实时出租车轨迹流数据, 以对实时的交通路况进行评测完成推荐. 其中出租车的 GPS 数据格式如表 1 所示. 实验使用 java 语言实现算法的编写, 并在 Windows 8.1 操作系统, 机器配置为 2.20 GHz Intel Core i5 处理器和 4 GB 物理内存的PC机上运行.

4.2 实验分析

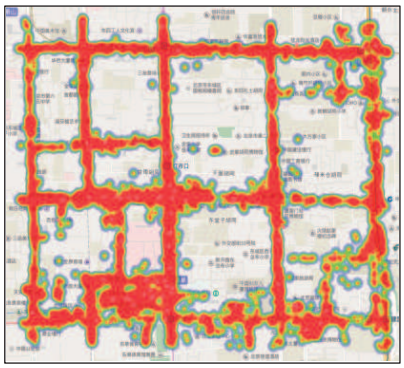
本文实验选取北京市王府井百货中心周围区域作为实验区域, 并提取该区域中的轨迹数据, 即满足东经度 lon$\in$ [116.405 467°, 116.436 109°]、北纬度 lat$\in$ [39.906 797°, 39.926 151°]. 基于以上实验区域, 根据本文中的载客热门区域生成方法, 在时空聚类过程中选择不同的参数, 并借助高德地图接口对载客热门区域进行可视化操作, 如图 5(b)、5(c) 所示.

表 1　出租车 GPS 数据格式

Tab. 1　Taxi GPS trajectory data format

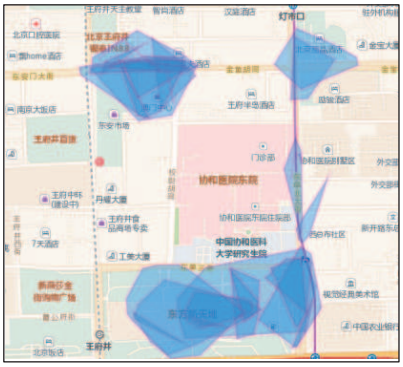
数据字段	数据类型	描述 (示例)
Time	char	时间信息 (20131023000013)
ID	char	出租车唯一标识 (001141)
Longitude	float	经度值 (116.439 972)
Latitude	float	纬度值 (39.850 876)
Speed	short	速度值 (62.000 000)
Direction	short	方向信息 (170.000 000)
Stat	char	载客状态(1) 空载状态 (0)

由于载客热门区域具有时段属性, 因此本文认为只要具有不同时段属性的载客热门区域就应该作为不同的个体存在. 因此图 5(b) 以及图 5(c) 中重叠的多边形则是由于不同的时段属性造成的, 不能将其视为同一个载客热门区域. 同时参数 $\varepsilon_{\text{spatio}}$ 的设置对聚类结果影响较明显, 若 $\varepsilon_{\text{spatio}}$ 值设置过大, 聚类过程对噪声数据不敏感, 导致时空簇中空间密度较小, 而区域范围变大, 如图 5(c). 本文认为载客热门区域作为载客事件发生较频繁的区域, 聚类结果即时空簇中载客点分布较密集的载客热门区域更具参考价值.



(a) **载客点热力图**
(a) The themodynamic diagram

MinPts = 20, $\varepsilon_{\text{temporal}}$ = 5 min, $\varepsilon_{\text{spatio}}$ = 50 m
(b) **载客热门区域**
(b) Hotspots



MinPts = 30, $\varepsilon_{\text{temporal}}$ = 5 min, $\varepsilon_{\text{spatio}}$ = 100 m
(c) **载客热门区域**
(c) Hostpots

图 5　载客热门区域可视化

Fig. 5　Hotspots visualization

本文作为首次考虑实际交通路况的推荐系统, 为了验证推荐算法的有效性, 考虑以下情形.

若某出租车位于位置 $P = (116.411\,39°, 39.910\,533°)$ 完成一次载客交易后提出推荐请求, 系统当前时间戳为 $T = 9:00$. 该系统的推荐结果为 k 个空载时间开销值较小的载客热门区域, 设置 $k = 3$, 则推荐结果如图 6 所示, 本系统将以"H_1、H_2、H_3"的排列次序对出租车司机进行推荐, 其中载客热门区域 H_1 与位置 P 之间的最短路径长约 0.5 km, 预计通行时间约 60 s; H_2 与位置 P 之间的最短路径长约 1.1 km, 预计通行时间约 100 s; H_3 与位置 P 之间的最短路径长约 0.8 km, 预计通行时间约 150 s. 虽然载客热门区域 H_2 的最短路径距离稍大于载客热门区域 H_3, 而由位置 P 沿最短路径驶向 H_3 将途经 3 个红绿灯路口, 时常造成路段拥堵, 但由于位置 P 到 H_2 的最短路径的路段实时交通状况良好, 其空载行驶时间开销值则越小, 因此其空载时间开销值则优于载客热门区域 H_3.

为验证推荐方法的准确性, 我们计算推荐结果 H_1, H_2, H_3 基于各时段的载客概率 (Probability of take a passeger), 如图 7、图 8、图 9.

载客热门区域作为该系统的推荐元素, 其基于当前时间戳的载客发生事件概率越大, 则说明系统推荐准确性将越高, 基于以上折线图, 我们发现载客热门区域 H_1, H_2, H_3 在时段 08:00—10:00 时其发生载客事件概率相较于其余时段都较高, 因此出租车司机将这些载客热门区域作为寻客目的地将更容易搭载到下一位乘客.

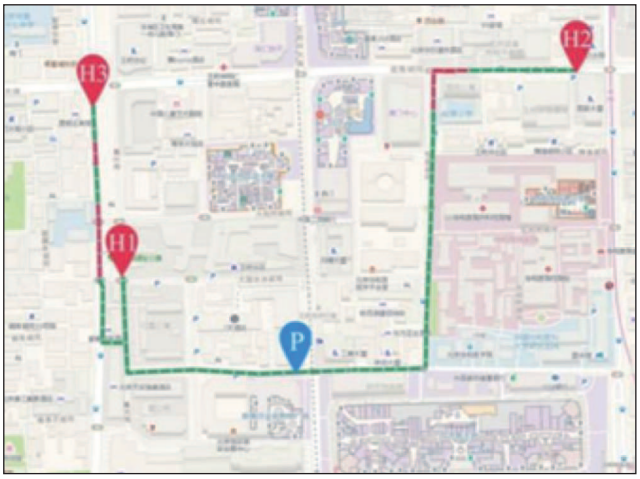


图 6 基于位置 P、时间 T 的载客热门区域 top-k 推荐可视化效果图

Fig. 6 The top-k hotspots for taxi based on P and T

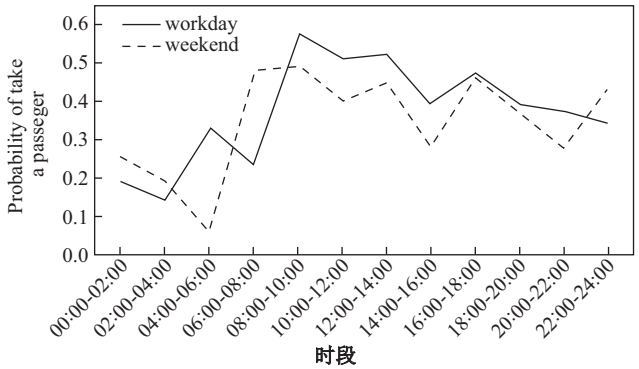


图 7 载客热门区域 H_1 的载客概率

Fig. 7 The probability of hotspot H_1

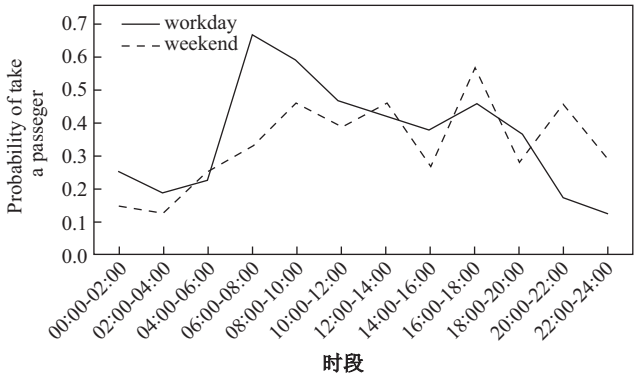


图 8　载客热门区域 H_2 的载客概率

Fig. 8　The probability of hotspot H_2

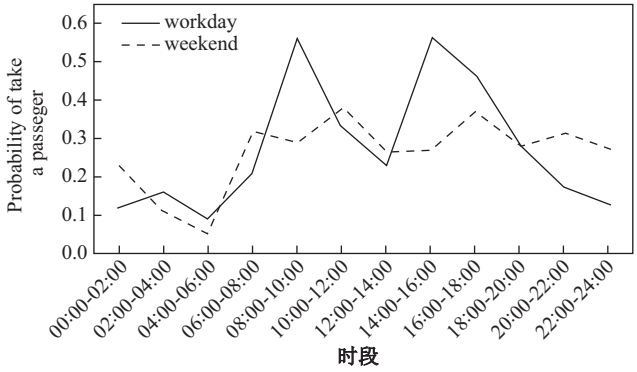


图 9　载客热门区域 H_3 的载客概率

Fig. 9　The probability of hotspot H_3

最后, 为验证该推荐方法的有效性, 将推荐结果 H_1, H_2, H_3 基于实际数据集的平均空载时间开销与传统推荐方法 (基于载客概率较大的推荐方法) 的推荐结果 H_1', H_2', H_3' 的平均空载时间开销进行比较. 提取出出租车历史轨迹数据集中当天处于 08:30—09:30 时间段内, 出租车在位置 P 的载客状态为空载状态, 且在较短时间内于预推荐的载客区域区域 H_1, H_2, H_3 内载客状态值跳变为载客状态的出租车轨迹. 并计算这些轨迹的平均空载时间开销. 同理可得与传统的推荐结果 H_1', H_2', H_3' 的平均空载时间开销. 如图 10 所示, 基于实时路况的载客热门区域推荐相较于传统的载客热门区域推荐方法, 因其主要考虑了实际的交通路况, 其平均空载时间开销优于传统的载客热门区域推荐方法.

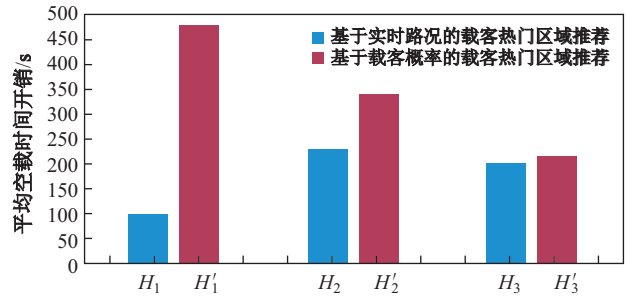


图 10　平均空载时间开销柱状图

Fig. 10　The histogram of average empty time cost

5 结 语

为解决出租车网约车空载现象严重的问题, 本文结合实时路况提出了一个top-k出租车载客热门区域的推荐方法, 该方法基于潜在空载时间开销较小的推荐策略向出租车司机进行实时推荐. 实验结果表明, 由于考虑了实际道路的交通情况, 能有效提升载客热门区域推荐的准确率.

[参 考 文 献]

[1] CHANG H W, TAI Y C, HSU Y J. Context-aware taxi demand hotspots prediction [J]. International Journal of Business Intelligence and Data Mining, 2010, 5(1): 3-18.

[2] LI X L, PAN G, WU Z H, et al. Prediction of urban human mobility using large-scale taxi traces and its applications [J]. Frontiers of Computer Science, 2012, 6(1): 111-121.

[3] QU M, ZHU H, LIU J, et al. A cost-effective recommender system for taxi drivers [C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2014: 45-54.

[4] YUAN N J, ZHENG Y, ZHANG L, et al. T-Finder: A recommender system for finding passengers and vacant taxis [J]. IEEE Transactions on Knowledge & Data Engineering, 2013, 25(10): 2390-2403.

[5] LI B, ZHANG D Q, SUN L, et al. Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset [C]// Proceedings of the Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on. IEEE, 2011: 63-68.

[6] MOREIRA-MATIAS L, GAMA J, FERREIRA M, et al. Predicting taxi–passenger demand using streaming data [J]. IEEE Transactions on Intelligent Transportation Systems, 2013, 14(3): 1393-1402.

[7] PHITHAKKITNUKOON S, VELOSO M, BENTO C, et al. Taxi-aware map: Identifying and predicting vacant taxis in the city [C]//AmI'10 Proceedings of the First International Joint Conference on Ambient Intelligence. 2010: 86-95.

[8] 齐观德, 潘遥, 李石坚, 等. 基于出租车轨迹数据挖掘的乘客候车时间预测 [J]. 软件学报, 2013, 24(2): 14-23.

[9] MAO J L, SONG Q G, JIN C Q, et al. TSCluWin: Trajectory stream clustering over sliding window [C]// DASFAA 2016: Database Systems for Advanced Applications. 2016: 133-148.

[10] WANG Z C, LU M, YUAN X R, et al. Visual traffic jam analysis based on trajectory data. [J]. IEEE Transactions on Visualization & Computer Graphics, 2013, 19(12): 2159-2168.

[11] HAN B, LIU L, OMIECINSKI E. Road-network aware trajectory clustering: Integrating locality, flow, and density [J]. IEEE Transactions on Mobile Computing, 2015, 14(2): 416-429.

[12] GRAHAM R L. An efficient algorith for determining the convex hull of a finite planar set [J]. Information Processing Letters, 1972, 4(1): 132-133.

[13] DIJKSTRA E D. A note on two problem in connexion with graphs [J]. Numerische Mathematik, 1959(1): 269-271.

[14] MITCHELL T, BUCHANAN B, DEJONG G, et al. Machine learning [J]. Annual Review of Computer Science, 1990(4): 417-433.

(责任编辑: 李 艺)