doi:10.3772/j.issn.1002-0470.2020.07.003

# 基于车牌时空数据的城市热点交通线路挖掘①

张翔宇②[*,**,****] 张 强[*,**] 吕明琪[***] 李素玲[*****]

([*]中国科学院计算技术研究所 北京 100190)

([**]中国科学院大学 北京 100049)

([***]浙江工业大学计算机学院 杭州 310014)

([****]北京赛迪时代信息产业股份有限公司 北京 100048)

([*****]中华全国总工会 北京 100085)

**摘 要** 交通摄像头在智能交通系统(ITS)中的作用日益重要,其主要功能为车牌识别。本文提出了一种从车牌时空数据中挖掘城市热点交通线路的方法。其中,车牌时空数据由部署在城市不同道路的交通摄像头不断进行车牌识别得到。实现该目标存在以下挑战:首先,一辆车的轨迹(由车牌时序数据代表)通常只占城市热点交通线路的一部分。其次,车牌识别存在高度的不确定性(如遗漏和错误),使得现有模式挖掘算法难以发现完整的城市热点交通线路。针对以上问题,本文提出了由2部分构成的方法。首先,该方法提出了一个基于子模式拼接的挖掘算法,从车牌时空数据中挖据出候选城市热点交通线路。然后,该方法基于一个聚类排序算法从候选城市热点交通线路中挑选出代表性城市热点交通线路。本文基于真实车牌时空数据对提出的方法进行了评测。

**关键词** 热点交通线路;交通摄像头;车牌时空数据;智能交通系统(ITS)

## 0 引 言

智能交通系统(intelligent transportation system, ITS)是改善城市交通系统运行性能的有效手段。近年来,越来越多的交通传感设备(如交通摄像头、环形线圈、微波检测器等)被部署在城市道路上,这些交通传感设备采集了大量的交通数据,使得智能交通系统逐渐从技术驱动为主演化为数据驱动为主[1]。因此,从交通大数据中挖掘交通运行模式成为了一个热门的研究领域。其中,城市热点交通线路(以下简称为"热点线路")是一类典型的交通运行模式,指在固定时间段内大量车辆共同行驶的道路路段序列[2,3]。热点线路可支持许多潜在的应用,如路线规划[4]、交通流预测[5]、拥堵预测[6]、城市规划[7]等。

然而,与单一车辆的行驶模式[8]不同,热点线路考虑的是城市车辆的总体流动规律。由于单一车辆行驶的不确定性,大部分车辆的行驶轨迹只贡献热点线路的一部分,只有极少数车辆的行驶轨迹能完整覆盖一条热点线路。因此,现有轨迹模式挖掘算法要求车辆轨迹完整覆盖行驶模式,则通常只能挖掘出很短的热点线路,对城市交通总体规划的指导意义不大。

针对热点线路挖掘问题,大多现有工作采用细粒度的车辆GPS轨迹数据[7,9-12]。然而,由于普通车辆的GPS轨迹数据基本无法获得,现有工作均采用浮动车辆(如出租车、公交车)的GPS轨迹数据。

① 国家自然科学基金联合重点项目(U1936215)资助。

② 男,1977年生,博士;研究方向:计算机系统结构,数据挖掘;联系人,E-mail: zhangxiangyu@qq.com

(收稿日期:2019-07-31)

由于浮动车辆的占比很小,其产生的轨迹数据对城市道路的时空覆盖非常有限。因此,从浮动车辆轨迹数据中挖掘出的热点线路通常难以反映真实的城市交通状况。另一方面,交通摄像头作为智能交通系统的重要基础设备,由于其非侵入的特性,已经被大量地部署在城市道路上,并被用作车辆行驶轨迹采集的最主流设备[1]。交通摄像头最主要的功能为车牌识别,由于交通摄像头具有空间属性,车牌识别具有时间属性,则一辆车的轨迹可被重构为该车辆按时间顺序经过的交通摄像头的序列[13]。鉴于交通摄像头的高覆盖率,从车牌识别时空数据中挖掘热点线路是更为合理的思路。

然而,基于车牌时空数据的热点线路挖掘比一般的交通模式挖掘更具挑战,原因如下:首先,由于建设成本的原因,交通摄像头通常无法覆盖所有的道路路段。其次,由于技术限制的原因,车牌识别经常存在遗漏和错误等问题。这些不确定性问题导致现有交通模式挖掘方法无法有效挖掘出热点线路,原因在于现有方法大多将车辆轨迹映射到道路网络上,在此基础上进行交通模式挖掘[2, 14-16]。然而,由于交通摄像头的空间稀疏性和识别不确定性,基于车牌时空数据重构得到的车辆轨迹中的连续 2 个交通摄像头可能间隔多个道路路段,导致难以进行道路匹配。另外,大部分车辆的行驶轨迹只占热点线路的部分。通常情况下,一辆车会在一条热点线路的起点和终点之间驶入和驶出,而大量车辆在该起点和终点之间的轨迹共同构成了这条热点线路。因此,现有交通模式挖掘算法(如聚类算法[17, 18]、序列模式挖掘算法[19, 20])只能挖掘出大量很短的热点线路,对城市交通总体规划的指导意义不大。交通模式挖掘算法通常会产生大量的挖掘结果,而这些结果中很多是相似和冗余的,导致决策人员难以从中发现最有价值的信息。

针对上述问题,本文提出了一种从车牌时空数据中有效挖掘热点线路的方法。该方法首先从重构的车辆轨迹数据中挖掘出子模式,并基于一个双向树数据结构拼接这些子模式以形成候选热点线路(该步骤称为热点线路挖掘)。然后采用聚类排序算法从候选热点线路中挑选出代表性热点线路(该步骤称为热点线路压缩)。本文的主要贡献如下。(1) 提出了一种无需道路网络支持的、基于车牌时空数据的热点线路挖掘方法。(2) 提出了一种双向树数据结构,用于拼接短的子模式以形成长的热点线路。(3) 提出了一种聚类排序算法,用于发现代表性热点线路。(4) 基于杭州市真实车牌时空数据进行了实验。

## 1 相关工作

现有工作主要采用数据挖掘技术从各类交通传感设备数据中挖掘交通模式。例如,Inoue 等人[15]和 Banaei-Kashani 等人[16]从环形线圈产生的交通流数据中挖掘每条道路的交通流模式。Yuan 等人[9]从浮动车辆轨迹数据中挖掘行驶模式,并用于最快路径检索服务。Zheng 等人[10]从浮动车辆轨迹数据挖掘两类出行模式即热点区域和热点路线,并基于这两类出行模式分析居民出行的时空规律。Janecek 等人[21]基于蜂窝网络数据对交通状态进行推断,包括行驶时间和交通拥堵等。

现有基于车牌时空数据挖掘的工作大多集中在交通流估计方面。例如,Castillo 等人[13]利用车牌时空数据和道路车流数据对车辆轨迹进行重构,在此基础上构建车辆轨迹矩阵。Mínguez 等人[22]对交通摄像头的数量和部署位置进行优化,在此基础上对 OD(起点-终点)矩阵进行估计。然而,这些方法粒度较粗,只能对某条道路上或某对起点终点间的交通流进行总体统计,无法估计车辆的细粒度行驶模式,从而无法支持热点线路的挖掘。

现有细粒度行驶模式挖掘方法大多基于细粒度的轨迹数据(如 GPS 轨迹数据)。例如,Yao 等人[17]提出了一个基于深度学习的轨迹聚类算法,首先提取时空不变特征,然后基于 seq2seq 模型对轨迹数据进行表征,最后在深度表征基础上实现聚类。Cao 等人[19]将行驶模式元素定义为频繁路段周围的区域,在此基础上提出了一种基于子串树数据结构的行驶模式挖据算法。然而,聚类算法或序列模式挖掘算法挖掘出的行驶模式通常较短,难以对热点线路进行有效表征,且这些工作挖掘出的基本上都是

单一对象的行驶模式。实际上，单一车辆的行驶模式通常只占热点线路的一小部分[2]。例如，城市中可能存在一条从居住区到工作区的热点线路，但通常大部分车辆不会行驶整条热点线路，而是在这条热点线路的中间某处驶入、某处驶出。

与本文工作最相关的是 Li 等人[2]提出的 FlowScan 算法。FlowScan 算法采用一个密度聚类算法在道路网络中对热点道路路段进行扩展，从而形成热点线路。然而，FlowScan 算法采用的是细粒度车辆轨迹数据，不适应车牌时空数据。首先，由于交通摄像头的空间稀疏性和识别不确定性，可能存在部分道路路段未部署交通摄像头，或未能正确识别部分行驶车辆的车牌，导致重构得到的车辆轨迹数据难以准确映射到道路网络中。其次，FlowScan 算法会产生大量的热点线路，其中某些是相似和冗余的，导致决策人员难以从中发现最有价值的信息。

## 2　问题定义

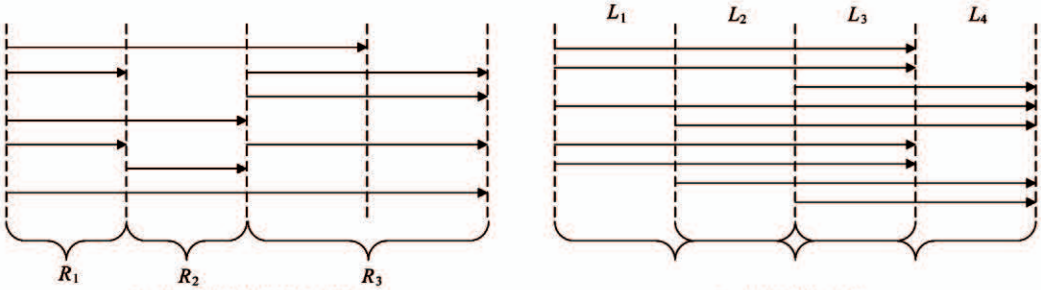
**定义 1**　车牌时空数据。交通摄像头可识别过往车辆的车牌并记录时间。因此，车牌时空数据可被定义为一个三元组的集合 $LD = \{(I_k, C_k, T_k)\}$，$I_k$ 为车辆 $k$ 的车牌号，$C_k$ 为记录 $I_k$ 的交通摄像头的编号，$T_k$ 为 $C_k$ 记录 $I_k$ 的时间。其中，$C_k$ 代表了空间属性（每个交通摄像头的经纬度位置已知），$T_k$ 代表了时间属性。

**定义 2**　车辆轨迹。对车牌时空数据进行交叉检索和重构可得到每辆车的轨迹。一辆车的轨迹可被定义为一个序列 $VT = <E_k>$，其中 $E_k$ 为一个二元组 $E_k = (C_k, T_k)(T_k < T_{k+1})$，$(C_k, T_k)$ 代表该车辆在时间 $T_k$ 行驶经过交通摄像头 $C_k$。在实际操作中，按天对车辆轨迹进行分割。

**定义 3**　热点线路。一条热点线路被定义为一个序列 $R = <C_k>$（$C_k$ 代表一个交通摄像头），其中 $R$ 中每连续 $K$ 个交通摄像头在指定时间段内共享的车辆轨迹超过 *minSup* 条。

## 3　热点线路挖掘

由于单一车辆行驶的不确定性，现有基于轨迹模式挖掘算法的热点线路挖掘方法通常只能挖掘出很短的热点线路。如图 1(a)所示，若要求每个行驶模式至少有 4 条轨迹支持，则现有轨迹模式挖掘算法只能挖掘出 $R_1$、$R_2$ 和 $R_3$ 这 3 条较短的热点线路。针对此问题，弱化现有轨迹模式挖掘算法的限制，将热点线路定义为一个道路路段的序列，该序列的指定长度的任意子序列均共享大量共同车流。如图 1(b)所示，$<L_1, L_2, L_3, L_4>$ 为一条热点线路，由于其长度为 2 的任意子序列（即 $<L_1, L>$、$<L_2, L_3>$ 和 $<L_3, L_4>$）均共享大于等于 5 条车辆轨迹。




(a) 轨迹模式挖掘算法的问题　　(b) 热点线路实例

**图 1　热点线路挖掘问题**

由于交通摄像头的空间稀疏性和识别不确定性，重构得到的车辆轨迹无法准确映射到道路网络。因此，本文提出了一种无需道路网络的从车牌时空数据中挖掘热点线路的方法，该方法分为子模式挖掘和子模式拼接 2 个步骤。

子模式挖掘工作流程如下，给定时间段 $[T_s, T_e]$，采用子串模式挖掘算法从车辆轨迹数据中挖掘出长度为 $K$ 的子串模式。子串和子序列是最具代表性的 2 类模式，而子串模式与子序列模式的不同在于，子串模式要求模式中连续的元素在原始序列

中也是连续的,而子序列模式只要求模式中连续的元素在原始序列中顺序一致(可以不连续)。之所以使用子串模式,是考虑到车辆在空间中的运动必须是连续的[19]。采用 N-Gram 算法挖掘子串模式:采用一个哈希表存储每个子串模式和其对应的支持度(即该子串模式出现的次数)。对每条车辆轨迹 $VT$,算法读入连续的 $K$ 个元素 $<(C_1, T_1), \cdots, (C_K, T_K)>$。如果 $T_1 \geqslant T_s$ 且 $T_K \leqslant T_e$(即轨迹发生在指定时间段内),则将 $<C_1, \cdots, C_K>$ 作为一个候选子串模式,并将该候选子串模式在哈希表中的支持度增加 1。最终,输出所有支持度大于等于 *minSup* 的候选子串模式。

实际操作中,核心参数 $K$ 和 *minSup* 的设置非常重要。一方面,参数 $K$ 设置的过大会导致难以发现足量的子模式,而设置的过小则对热点线路的要求过低(容易导致产生过多无意义的热点线路)。另一方面,参数 *minSup* 的设置需要考虑实际的交通流量。在交通流量较大的区域中,*minSup* 也应设置的较大。然而,由于不同区域的实际交通流量差异较大,导致 *minSup* 的绝对数值难以统一设置。因此,通过设置一个(0, 1)的相对数值来估算 *minSup* 的绝对数值,方法如下:首先,提取所有长度为 2 的子模式并计算它们的支持度。然后,绘制所有长度为 2 的子模式支持度的 CDF(累积分布函数)曲线。最后,设置一个 *minSup* 的相对数值 *rMinSup*(0 < *rMinSup* < 1),并基于式(1)计算 *minSup* 的绝对数值(*minSup* 的绝对数值为 *iCDF*(*rMinSup*),*iCDF*(*)为累积分布反函数)。

$$iCDF(p) = \inf\{x \in R : p \leqslant CDF(x)\} \quad (1)$$

子模式拼接工作流程如下。提出了一种将短的子模式拼接成长的热点线路的算法。算法流程如算法 1 所示,其主要工作为基于前向拼接和后向拼接概念构造一棵双向树。首先,基于一个种子子模式(第 3 行),算法调用递归函数不断将其他子模式拼接到种子子模式上(第 6 行),形成一棵双向树(包括一棵前向树和一棵后向树)。然后,拼接各前向树分支和各后向树分支,得到候选热点线路(第 7 ~ 9 行)。

**定义 4** 前向拼接和后向拼接。给定一个长度为 $K$ 的子模式 $P_0$,另一个长度为 $K$ 的子模式 $P_1$ 若满足:$P_1$ 的长度为 $K-1$ 的前缀可与 $P_0$ 的长度为 $K-1$ 的后缀完全匹配,则 $P_1$ 可与 $P_0$ 前向拼接。另一个长度为 $K$ 的子模式 $P_2$ 若满足:$P_2$ 的长度为 $K-1$ 的后缀可与 $P_0$ 的长度为 $K-1$ 的前缀完全匹配,则 $P_2$ 可与 $P_0$ 后向拼接。

**算法 1** 子模式拼接算法

输入:长度为 $K$ 的子模式集合 *PS*

输出:候选热点线路集合 *RS*

1. 将 *PS* 复制到 *TS*
2. **while** *TS* 不为空 **do**
3. 从 *TS* 中找出支持度最大的子模式 *P*
4. 将 *P* 从 *TS* 中删除
5. 构建 2 个树节点 *fn*(对应 $P.C_K$)和 *bn*(对应 $P.C_1$)
6. 运行函数 Forward _ Expand(*fn*)和 Backward _ Expand(*bn*)
7. **for** 以 *bn* 为根节点的后向树的每个分支 *bb* **do**
8. **for** 以 *fn* 为根节点的前向树的每个分支 *fb* **do**
9. 将 *bb* 的反转,$<P.C_2, \cdots, P.C_{K-1}>$ 和 *fb* 进行拼接,得到 *R*,并将 *R* 加入 *RS*

函数 **Forward _ Expand**(树节点 *n*)

1. 设 *P* 为树节点 *n* 对应的子模式
2. 从 *PS* 中找出所有可与 *P* 前向拼接的子模式集合 *FS*
3. **for** *FS* 中每个子模式 *fp* **do**
4. 构建一个新的树节点 *nn*(对应 $fp.C_K$),并将其插入为树节点 *n* 的子节点
5. 将 *fp* 从 *TS* 中删除,并运行函数 Forward _ Expand(*nn*)

算法 1 续

函数 **Backward _ Expand**(树节点 $n$)

1. 设 $P$ 为树节点 $n$ 对应的子模式
2. 从 $PS$ 中找出所有可与 $P$ 后向拼接的子模式集合 $BS$
3. **for** $BS$ 中每个子模式 $bp$ **do**
4. 构建一个新的树节点 $nn$(对应 $bp.C_1$),并将其插入为树节点 $n$ 的子节点
5. 将 $bp$ 从 $TS$ 中删除,并运行函数 Backward _ Expand($nn$)

下面给出一个实例对子模式拼接算法进行进一步说明。给定 7 个长度为 3 的子模式 $P_1$ = <1, 2, 3>,$P_2$ = <2, 3, 4>,$P_3$ = <2, 3, 5>,$P_4$ = <3, 4, 6>,$P_5$ = <3, 4, 7>,$P_6$ = <8, 1, 2> 和 $P_7$ = <9, 1, 2>,其中 $P_1$ 的支持度最大,则构造的双向树如图 2 所示(其中,树节点用圆形代表,而正方形代表的是树节点对应的子模式)。从该双向树中,通过拼接前向树分支和后向树分支,可以提取出 6 个候选热点线路,即 $R_1$ = <8, 1, 2, 3, 5>,$R_2$ = <9, 1, 2, 3, 5>,$R_3$ = <8, 1, 2, 3, 4, 6>,$R_4$ = <8, 1, 2, 3, 4, 7>,$R_5$ = <9, 1, 2, 3, 4, 6> 和 $R_6$ = <9, 1, 2, 3, 4, 7>。

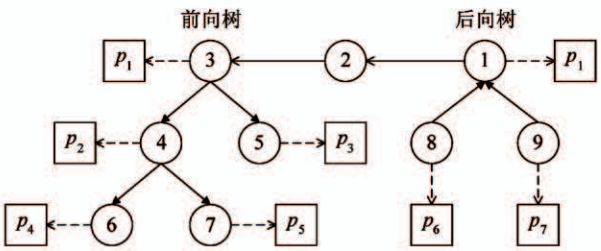



**图 2 子模式拼接实例**

## 4 热点线路压缩

热点线路挖掘步骤会产生大量候选热点线路,其中存在大量相似和冗余,导致决策人员难以从中发现最有价值的信息。以图 2 为例,仅 7 个子模式就会产生 6 个候选热点线路,而真实交通数据中通常可挖掘出海量的子模式,导致产生的候选热点线路过多。此外,从图 2 中产生的候选热点线路中可发现,许多候选热点线路非常相似(如 $R_1$ 和 $R_2$,$R_3$ 和 $R_4$,$R_5$ 和 $R_6$)。针对此问题,提出了一个聚类排序算法,用于对候选热点线路进行压缩,主要包括聚类和排序 2 个步骤:首先,聚类步骤对所有候选热点线路进行聚类。然后,排序步骤从每个聚类中挑选出最具代表性的候选热点线路。

在聚类步骤中,基于最长公共子序列算法定义候选热点线路间的相似度,候选热点线路 $R_i$ 和 $R_j$ 的相似度计算方法如式(2)所示。

$$S(R_i, R_j) = \max\{S(R_i \rightarrow R_j), S(R_j \rightarrow R_i)\} \tag{2}$$

$$S(R_i \rightarrow R_j) = \frac{|LCSS(R_i, R_j)|}{|R_i|} \tag{3}$$

其中,$LCSS(R_i, R_j)$ 为 $R_i$ 和 $R_j$ 的最长公共子序列。式(2)中使用了 $S(R_i \rightarrow R_j)$ 和 $S(R_j \rightarrow R_i)$ 的最大值作为 $R_i$ 和 $R_j$ 的相似度,其目的是为了更有利于较长的候选热点线路。这种情况下,较长的候选热点线路更容易与其他的候选热点线路产生高相似度,使得较长的候选热点线路倾向于将较短的相似候选热点线路吸收进同一聚类,并更容易被挑选为聚类中的代表性候选热点线路。

在此基础上,采用 Affinity Propagation 聚类算法[23]对所有候选热点线路进行聚类处理。Affinity Propagation 聚类算法的优势在于不需要预先设定聚类的数量,这符合热点线路数量通常未知的现实情况。

在排序步骤中,从每个聚类中挑选出最具代表性的候选热点线路。为此,给定一个聚类,计算其中每个候选热点线路的权值,而热点线路权值由表征度权值和重要度权值 2 部分构成。

其中,表征度权值用于指示一个候选热点线路代表其他候选热点线路的能力,而候选热点线路 $R_j$ 代表候选热点线路 $R_i$ 的能力可由 $S(R_i \rightarrow R_j)$ 判断,即 $R_i$ 和 $R_j$ 的最长公共子序列能较多覆盖 $R_i$。因此,基于式(4)对候选热点线路 $R_k$ 的在聚类 $RC$ 中的表征度权值进行量化。

$$RScore_{RC}(R_k) = \frac{\sum_{R_i \in RC} S(R_i \rightarrow R_k)}{|RC|} \tag{4}$$

另一方面,重要度权值用于指示一个候选热点线路是否经过城市的重要区域。综合考虑以下假设对候选热点线路的重要度权值进行量化。(1) 如果一个候选热点线路包含的交通摄像头更重要,则该候选热点线路更重要。(2) 如果一个交通摄像头被包含在更重要的候选热点线路中,则该交通摄像头更重要。(3) 一个交通摄像头的重要度可由其检测到的交通流量进行量化。

上述假设中,假设(1)和假设(2)表明候选热点线路和交通摄像头间存在互增强关系。这与网页排序算法 HITS 中定义的 hub 页和 authority 页间的关系类似,而 HITS 算法正是利用这种互增强关系计算网页的重要度权值。因此,将交通摄像头看成 hub 页,将候选热点线路看成 authority 页,然后基于 HITS 算法的思想计算候选热点线路的重要度权值。给定 $n$ 个交通摄像头和 $m$ 个候选热点线路,构建一个 $m \times n$ 的矩阵 $\boldsymbol{M}_{RC}$(其中 $\boldsymbol{M}_{RC}[i, j]$指示第 $i$ 条候选热点线路是否包含第 $j$ 个交通摄像头)。假定 $P_C$ 和 $P_R$ 分别代表 hub 分数向量和 authority 分数向量,则可采用幂迭代算法计算最终的 $P_C$ 和 $P_R$。为将假设(3)反映在算法里,在每轮幂迭代中,authority 分数(代表候选热点线路的分数)会按照交通摄像头检测到的交通流量成比例地传播到 hub 分数(代表交通摄像头的分数)中,即交通摄像头检测到的交通流量作为分数传播的一个因子。算法 2 展示了基于幂迭代的候选热点线路重要度权值计算算法,其中 $\boldsymbol{V}_T$ 为一个 $n$ 维的列向量($\boldsymbol{V}_T[k]$代表第 $k$ 个交通摄像头历史上在指定时间段内检测到的平均交通流量)。

图 3 给出了一个候选热点线路权值计算的实例。给定 3 条候选热点线路($R_1$, $R_2$ 和 $R_3$)和 4 个交通摄像头($C_1$, $C_2$, $C_3$ 和 $C_4$),假定 $R_1$, $R_2$ 和 $R_3$ 属于同一聚类,则 $R_1$, $R_2$ 和 $R_3$ 的表征度权值分别为 0.833,0.5 和 0.583。而当执行候选热点线路重要度权值计算算法后,$R_1$, $R_2$ 和 $R_3$ 的重要度权值分别为 0.363,0.248 和 0.389。因此,$R_1$ 具有最高的表征度权值(由于 $R_1$ 与其他候选热点线路的交叠部分最多),而 $R_3$ 具有最高的重要度权值(由于 $R_3$ 经过最重要的交通摄像头 $C_4$)。

**算法 2** 候选热点线路重要度权值计算算法

输入:关联矩阵 $\boldsymbol{M}_{RC}$,交通流量向量 $\boldsymbol{V}_T$

输出:所有候选热点线路的重要度权值

1. 初始化 $P_C^{(0)} = [\underbrace{1,1,\cdots,1}_{n}]^{\mathrm{T}}$, $P_R^{(0)} = [\underbrace{1,1,\cdots,1}_{m}]^{\mathrm{T}}$
2. **while** 算法未收敛 **do**
3. $P_C^{(t+1)} = \boldsymbol{M}_{RC}^T \cdot P_R^{(t)} \cdot \boldsymbol{V}_T$, $P_R^{(t+1)} = \boldsymbol{M}_{RC} \cdot P_C^{(t+1)}$ // 第 $t$ 轮迭代
4. 将 $P_R^{(t+1)}$ 归一化
5. 将最终 $P_R$ 输出为重要度权值向量

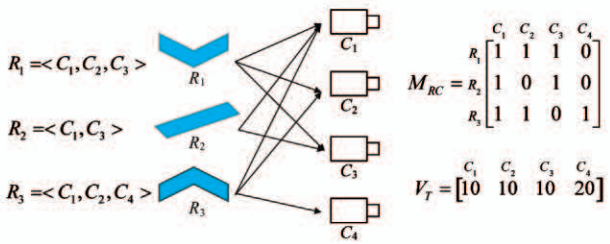



**图 3** 一个候选热点线路权值计算的实例

得到聚类中每个候选热点线路的表征度权值 $rs$ 和重要度权值 $is$ 后,可对 2 个权值进行加权求和得到候选热点线路权值 $ws$(如式(5)所示),然后从每个聚类中挑选出权值最高的候选热点线路作为最终的热点线路,而这些最终的热点线路可按照其包含的车流量进行倒排排序。

$$ws = \alpha \times rs + (1 - \alpha) \times is \tag{5}$$

## 5 实 验

### 5.1 数据集

采用杭州市的真实车牌时空数据集进行实验,该数据集包含了部署在杭州市区的 821 个交通摄像头,数据集时间跨度为 2012 年 6 月 1 日至 2012 年 7 月 4 日,以及 2018 年 11 月 5 日至 2018 年 12 月 1 日。实验仅考虑工作日(总共 44 d)。最终数据集包含了 195 579 733 个车牌识别记录,数据集中每天平均车牌识别记录数为 4 444 993(标准差为 457 313),每天平均检测到的车辆数为 724 636(标准差为 73 096)。

基于对该数据集的统计分析，发现交通摄像头存在较高的空间稀疏性和识别不确定性。如图 4 所示，当缩小地图后，可以发现有很多道路未部署交通摄像头。此外，该数据集中存在 23 365 567 条车牌识别记录是错误的（大概占总车牌识别记录数的 12%）。再次，交通摄像头还有可能未检测到经过的车辆。

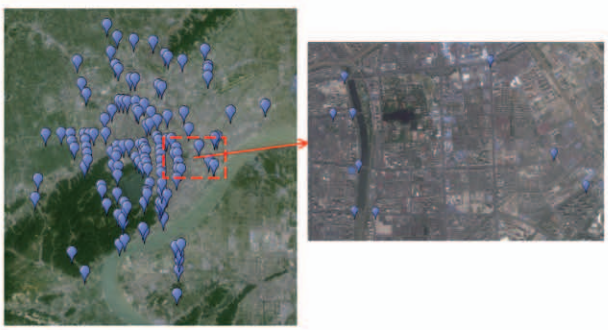


图 4　交通摄像头的空间稀疏性

### 5.2　调参实验

本文提出的方法包括 2 个核心参数为 $K$ 和 *minSup*，这 2 个核心参数的设置直接影响方法的性能。其中，参数 $K$ 必须大于等于 3，这是由于长度小于 3 的子模式无法被拼接（由于交通摄像头基本都部署在道路路口，因此 2 个交通摄像头构成一条道路路段，而长度为 2 的子模式只包含一条道路路段）。另一方面，参数 *minSup* 的设置需要考虑实际的交通流量。

在以下实验中，将时间段设置为 08:00 – 09:00（即早高峰）。图 5 显示了调整参数 $K$ 和 *rMinSup* 后方法性能的变化，这里方法性能的考查指标包括挖掘出的热点线路的平均长度、平均车流量、最小车流量和数量。其中，将一条热点线路包含的车流量计算为其包含的长度为 2 的子模式的平均支持度。通常情况下，希望挖掘出的热点线路的平均长度更长、车流量更大。如图5（a）和图5（b）所示，增大$K$和

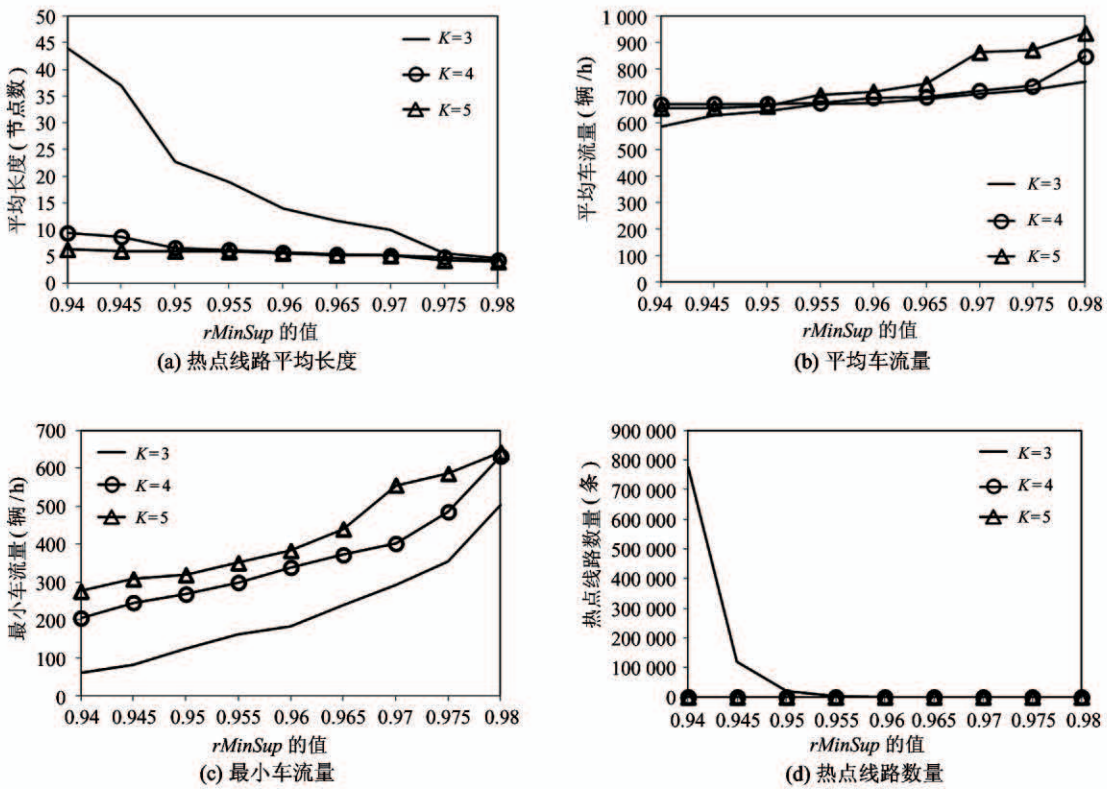



图 5　参数 $K$ 和 *rMinSup* 对方法性能的影响

*rMinSup* 后，热点线路平均长度显著缩短，而平均车流量少量增加。这说明 $K$ 和 *rMinSup* 应该设置的小一些。然而，将 $K$ 和 *rMinSup* 设置的过小会引发以下问题：挖掘出的热点线路包含的车流量过小（如图 5（c）所示）或挖掘出的热点线路数量爆炸性增加（如图 5（d）所示）。因此，将这 2 个参数设置如下：

$K=3$,$rMinSup=0.955$(对应的 $minSup$ 绝对数值为172)。

### 5.3 方法评测

将本文提出的方法(称为 OurMining)与以下方法进行比较评测。

(1) CloSpan。基于 CloSpan 算法[24]挖掘子模式,并基于第4节提出的方法对子模式进行压缩得到最终的热点线路。

(2) FlowScan。基于 FlowScan 算法[2]挖掘候选热点线路,并基于第4节提出的方法对候选热点线路进行压缩得到最终的热点线路。FlowScan 采用一个密度聚类算法在道路网络中将道路路段扩展到其邻接道路路段以形成热点线路。道路路段 $r$ 的邻接道路路段集 $RS$ 满足:$r$ 与 $RS$ 中任一道路路段间的最短道路网络距离小于 $Eps$。然而,基于车牌时空数据重构得到的车辆轨迹无法进行道路匹配。因此,将邻接道路路段概念修改为邻接交通摄像头概念,即交通摄像头 $c$ 的邻接交通摄像头集合 $CS$ 满足:$c$ 与 $CS$ 中任一交通摄像头的直线距离小于 $dEps$。在此基础上,采用 FlowScan 算法将交通摄像头扩展到其邻接交通摄像头以形成热点线路。

(3) DirectMining。基于第3节提出的方法挖掘候选热点线路,不对候选热点线路进行压缩,而仅基于包含的车流量对候选热点线路进行简单倒排排序。

设置 $\alpha = 0.5$,并将热点线路的平均长度和最大长度、top-$N$ 热点线路的城市车流覆盖率作为评价指标。其中,top-$N$ 热点线路的城市车流覆盖率为排序最靠前的 $N$ 个热点线路所包含的车流量(在本文中,即排序最靠前的 $N$ 个热点线路所包含的长度为2的子模式的支持度总和)占城市总车流量(在本文中,即所有长度为2的子模式的支持度总和)的比例。参数设置如下:$K = 3$(针对 OurMining 和 DirectMining),$rMinSup=0.955$(针对 OurMining,CloSpan,FlowScan 和 DirectMining),$dEps=1\,000 \sim 5\,000$ m(针对 FlowScan,对应 $FlowScan\_1000 \sim FlowScan\_5000$)。此外,为保证子模式元素的空间连续性,将 CloSpan 算法中模式元素的最大间隔限制为3。

实验结果如图6所示,可以发现以下现象。第1,CloSpan 仅能挖掘出很短的热点线路,这是由于序列模式挖掘算法要求支持某个序列模式的车辆轨迹必须完全覆盖这个序列模式,而通常一辆车的轨迹只能覆盖热点线路的一部分。第2,DirectMining 能够挖掘出最长的热点线路,但其 top-$N$ 热点线路的城市车流覆盖率较低(即使 $N$ 较大时)。这是由于当不进行热点线路压缩时,大量热点线路存在高度的重叠,特别是排序靠前的热点线路(由于这些热点线路基本上都集中在城市最繁忙的一、两条线路上),因此增大 $N$ 也无法提升 top-$N$ 热点线路的城市车流覆盖率。第3,当 $dEps$ 设置的较低时,FlowScan 的性能较差,且 FlowScan 的 top-$N$ 热点线路的城市车流覆盖率低于 OurMining,特别是 $N$ 较大时。对实验结果进行分析后发现,FlowScan 可发现高质量的热点线路,但其召回率较低,这主要是由于邻接交通摄像头的不确定性造成的。首先,基于直线距离定义邻接关系不合理(原始 FlowScan 是基于道路网络距离定义道路路段的邻接关系)。例如,同样是部署在一条道路路段两端的交通摄像头,当该道路路段是一条城市高速路时,其直线距离会很大,导致算法认为其不存在邻接关系。其次,FlowScan 的密度聚类算法不适应车牌时空数据。FlowScan 的密度聚类算法挑选初始种子道路路段的方式为该道路路段包含至少 $minSup$ 起始车流量或 $minSup$ 终止车流量,而后者在车牌时空数据中无法考查。综上,OurMining 可发现质量更高的热点线路(长度更长,且更少的实例就能覆盖更多的城市车流)。

图7展示了2条挖掘出的热点线路(图7(a)展示了从早高峰时间段中挖掘出的排名第1的热点线路,而图7(b)展示了从晚高峰时间段中挖掘出的排名第1的热点线路)。图中热点线路由一个折线代表(其中,圆点代表起点,箭头代表终点),2个交通摄像头之间的折线为其在道路网络中的最短路径。由图7可以看出,杭州市早高峰最繁忙的热点线路为由城市北部去往城市中心,而晚高峰最繁忙的热点线路为由滨江区穿越城市中心高架路去往城市西部居住区。

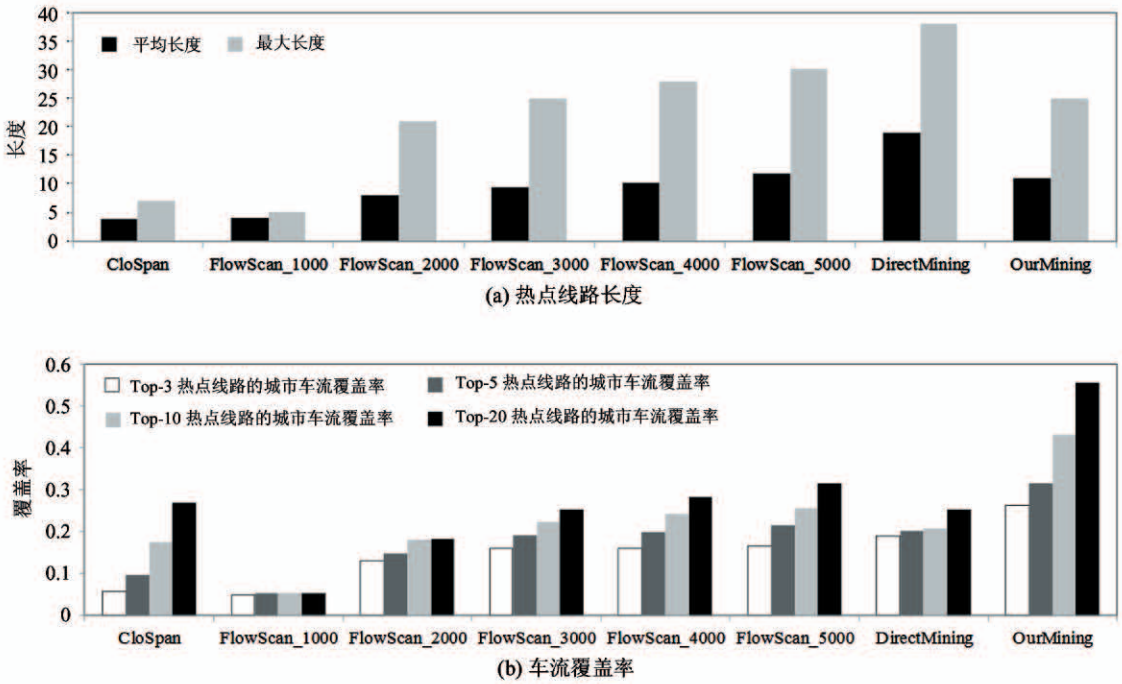



图 6　不同方法的性能比较


图 7　热点线路可视化

## 6　结　论

本文提出了一种从车牌时空数据中挖掘热点线路的方法。该方法首先挖掘和拼接子模式以形成候选热点线路，然后对候选热点线路进行聚类和排序以得到代表性热点线路。基于杭州市真实车牌时空数据的实验结果表明：与现有方法相比，本文提出的方法可以发现更有价值的热点线路（长度更长，且较少的实例就能覆盖较多的城市车流）。

**参考文献**

[1] Zhang J, Wang F Y, Wang K, et al. Data-driven intelligent transportation systems: a survey[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2011, 12(4): 1624-1639

[2] Li X, Han J, Lee J G, et al. Traffic density-based discovery of hot routes in road networks[C]//Proceedings of the International Symposium on Spatial and Temporal Da-

tabases, Boston, USA, 2007: 441-459

[3] Cui C, Zheng L, Sun D. Mining private vehicle hot routes using electronic registration identification data[C]//Proceedings of the International Conference on Big Data Engineering, Shanghai, China, 2019: 51-56

[4] Almeida A M R, Leite J L A, Macedo J A F, et al. GPS2GR: optimized urban green routes based on GPS trajectories[C]//Proceedings of the ACM SIGSPATIAL Workshop on GeoStreaming, Redondo Beach, USA, 2017: 39-48

[5] Zhang J, Zheng Y, Qi D, et al. Predicting citywide crowd flows using deep spatio-temporal residual networks [J]. *Artificial Intelligence*, 2018, 259: 147-166

[6] Kong X, Xu Z, Shen G, et al. Urban traffic congestion estimation and prediction based onfloating car trajectory data [J]. *Future Generation Computer Systems*, 2016, 61: 97-107

[7] 肖露艳. 基于出租车轨迹数据的城市夜间公交线路规划研究[D]. 深圳: 中国科学院大学中国科学院深圳先进技术研究院, 2017: 28-43

[8] 赵晓光. 基于多尺度区域划分和运动模式的车辆轨迹预测[D]. 北京: 北京邮电大学信息与通信工程学院, 2017: 33-45

[9] Yuan J, Zheng Y, Xie X, et al. T-Drive: enhancing driving directions with taxi drivers' intelligence [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2013, 25(1): 220-232

[10] Zheng L, Xia D, Zhao X, et al. Spatial-temporal travel pattern mining using massive taxi trajectory data [J]. *Physica A: Statistical Mechanics and its Applications*, 2018, 501(1): 24-41

[11] 程智源. 基于轨迹聚类的交通热点分析[D]. 成都: 电子科技大学电子与通信工程学院, 2018: 36-47

[12] An S, Yang H, Wang J, et al. Mining urban recurrent congestion evolution patterns from GPS-equipped vehicle mobility data[J]. *Information Sciences*, 2016, 373(10): 515-526

[13] Castillo E, Menéndez J M, Jiménez P. Trip matrix and path flow reconstruction and estimation based on plate scanning and link observations [J]. *Transportation Research Part B: Methodological*, 2008, 42(5): 455-481

[14] 周晓云. 基于多尺度卷积神经网络的出行目的地预测技术研究[D]. 北京: 北京邮电大学软件学院, 2019: 21-32

[15] Inoue R, Miyashita A, Sugita M. Mining spatio-temporal patterns of congested traffic in urban areas from traffic sensor data[C]//Proceedings of the IEEE 19th Conference on Intelligent Transportation Systems, Rio de Janeiro, Brazil, 2016:731-736

[16] Banaei-Kashani F, Shahabi C, Pan B. Discovering patterns in traffic sensor data[C]//Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming, Chicago, USA, 2011: 10-16

[17] Yao D, Zhang C, Zhu Z, et al. Trajectory clustering via deep representation learning[C]//Proceedings of the International Conference on Neural Networks, Anchorage, Alaska, 2017: 3880-3887

[18] Yuan G, Sun P, Zhao J, et al. A review of moving object trajectory clustering algorithms[J]. *Artificial Intelligence Review*, 2017, 47(1): 123-144

[19] Cao H, Mamoulis N, Cheung D W. Discovery of periodic patterns in spatio temporal sequences [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2007, 19(4): 453-467

[20] Lipan F, Groza A. Mining traffic patterns from public transportation GPS data[C]//Proceedings of the IEEE 6th Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, 2013: 123-126

[21] Janecek A, Hummel K A, Valerio D, et al. Cellular data meet vehicular traffic theory: location area updates and cell transitions for travel time estimation[C]//Proceedings of the ACM Conference on Ubiquitous Computing, Pittsburgh, USA, 2012: 361-370

[22] Mínguez R, Sánchez-Cambronero S, Castillo E, et al. Optimal traffic plate scanning location for OD trip matrix and route estimation in road networks [J]. *Transportation Research Part B: Methodological*, 2010, 44(2): 282-298

[23] Frey B J, Dueck D. Clustering by passing messages between data points[J]. *Science*, 2007, 315(5814): 972-976

[24] Yan X, Han J, Afshar R. CloSpan: mining closed sequential patterns in large datasets[C]//Proceedings of the SIAM International Conference on Data Mining, San Francisco, USA, 2003: 166-177

# Mining urban hot routes based on spatio-temporal license plate number data

Zhang Xiangyu[*,**,****], Zhang Qiang[*,**], Lv Mingqi[***], Li Suling[*****]

([*] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([**] University of Chinese Academy of Sciences, Beijing 100049)

([***] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310014)

([****] Beijing CCID Info Tech Inc, Beijing 100048)

([*****] All-China Federation of Trade Unions, Beijing 100085)

**Abstract**

Traffic camera plays an important role in intelligent transportation systems (ITS). A major function of traffic camera is license plate number recognition. This paper focuses on discovering city-wide hot routes using license plate number data recorded by traffic cameras deployed throughout the city. This task is challenging due to the following two reasons: First, a vehicle trajectory could usually contribute to only a small portion of a hot route. Second, the high degree of uncertainty of license plate number data makes the existing mining algorithms ineffective. Aiming at these problems, a two-phase method is proposed. First, it extracts hot routes by aggregating the license plate number data from multiple traffic cameras and vehicles. Second, it compresses the mined hot routes based on a clustering and ranking algorithm. The proposed method is evaluated based on real-world license plate number data from a city-wide traffic camera system.

**Key words**: hot route, traffic camera, license plate number data, intelligent transportation systems (ITS)