文档类别

杭州海康威视数字技术股份有限公司

文档编号

一种基于朴素贝叶斯的交通轨迹预测方案

编　　制	辛杰
审　　批	

密级级别：[内部公开]
生效时间： 2019 年 2 月 28 日
保密期：无

目录

1. 简介

1.1 编写目的

本文依据车辆目前前进轨迹，对其在未来一段时间（一小时）之内最有可能经过的三个卡口进行预测，主要采用贝叶斯算法、以及改进的轻量级贝叶斯算法对问题进行算法实现与模型求解。本文主要提供轨迹预处理、前置轨迹截取长度、测试集的构造、候选集的选择以及时间特征的应用等几个方面的流程与经验，供相关工作参考。

1.2 背景

目前，车辆数量呈指数形式激增，对车辆行驶行为及其行驶偏好进行研究，不管是对于车流的控制还是非法车辆的追踪，都具有重要意义。针对这一问题卡口信息是不可或缺的因素，而这恰恰是我们的优势，基于卡口信息，可以恰当地描述车辆的前进轨迹，配合常规的路径数据挖掘算法，根据车辆前段时间行进轨迹预测其前方将要经过卡口，进一步挖掘出该车辆的行驶习惯，在此基础上实现其他应用。

1.3 工作说明

基于已有的车辆行驶轨迹记录，提取经过卡口的 id 与时间，分别采用贝叶斯算法、轻量级贝叶斯算法对车辆某一时刻的运行轨迹进行预测，获得其接下来一小时内到达概率最大的三个卡口 top3，然后比对其未来一小时实际经过卡口，测试预测算法的精确度。

测试时，基于当前卡口判断车辆的未来三个卡口，如果跟 top3 有交集，即视为预测准确，否则不准确。判断整个模型的预测准确性表示为：

$$\text{precision} = \frac{n}{N} * 100\%$$

这里，N 表示整个测试集的轨迹数，n 表示真实值在预测的 top3 中的轨迹数。

2. 方案设计

2.1 方案总体说明

该方案包括前期对数据的处理，贝叶斯模型的实现，后期对贝叶斯模型的优化改进，

以及对模型预测结果的评估。

2.2 数据清洗与预处理

这里，我们主要做包括不正确车牌、重复记录等的进行清洗以及预处理。

2.2.1 车牌清洗

车牌清洗，主要包括去除车牌识别不正确的车牌，如过滤掉默认“车牌”等的记录，具体规则细节详见附件 1。

2.2.2 轨迹拉通

进行聚合轨迹，这里主要指按车牌+车牌颜色，进行 groupby，并按过车时间进行升序排列。

2.2.3 重复记录清洗

记录去重，这里主要指对同一车牌在同一段时间范围内经过同一个卡口（这里同一段时间范围暂定 5 分钟）进行去重。

2.3 轨迹分割

目前我们拥有的数据是一段时间内汽车所有经过卡口记录按照车牌、以时间顺序排列的数据。而一天甚至数月之间，汽车并不是只进行一次出行，多次出行可能会被连接在一起。所以我们需要对记录进行拆分，获得独立的路径数据。

一条路径应当有起点、有终点。起点与终点之间的记录，应当是为了靠近终点而出现的经过卡口行为。这样的路径才可以说前后具有关联性，便于计算。如下图所示：

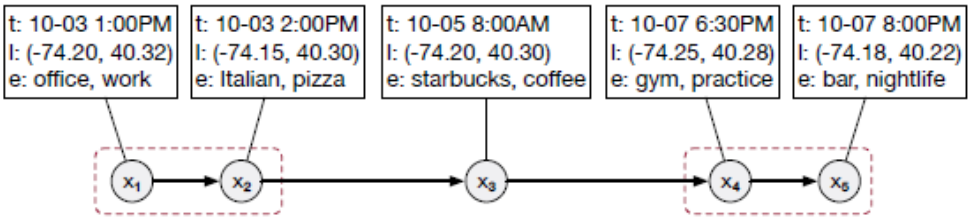

图 1 轨迹示意图

对一条卡口经过序列，若两次经过卡口的时间间隔大于一定阈值（经过探索数据初步

暂定为 2 小时），可以视为两次经过卡口之间路径有了中断，将该序列切割为两条序列，如下所示:

$$car:(8:00am,A),(9:00am,B),(13:00pm,C),(14:00pm,D)$$
$$\rightarrow\begin{cases} car:(8:00am,A),(9:00am,B) \\ car:(13:00pm,C),(14:00pm,D) \end{cases}$$

这一分割得到的数据适用于后续的算法方案。

2.4 算法方案

2.4.1 算法原理

朴素贝叶斯模型是基于贝叶斯原理的算法。贝叶斯原理中，假设是各项条件互相之间无关联，数学公式可以表示为： $p(c|x,y)=\dfrac{p(x,y|c)\times p(c)}{p(x,y)}$ 。

这里的 c 表示类别，输入待判断数据，式子给出要求解的某一类的概率。我们的最终目的是比较各类别的概率值大小，而上面式子的分母是不变的，因此只要计算分子即可。

2.4.2 算法应用

输入：车辆 id，（卡口 id，经过时间）序列，如（1）：

$$\begin{matrix} car_1 & (id,t) & \cdots & (id,t) \\ car_2 & (id,t) & \cdots & (id,t) \\ \vdots & \vdots & \ddots & \vdots \\ car_n & (id,t) & \cdots & (id,t) \end{matrix} \quad (1)$$

预测： $car_i \underbrace{(id_{first},t_{first})\rightarrow(id,t)\rightarrow\cdots\rightarrow(id_{last},t_{last})}_{Track_p}$ 在未来一段时间（如 1 小时）内可能出现在哪些卡口。

方案：贝叶斯公式（省略分母） $P(id_i|T)=P(id_i)P(T|id_i)$

$$P(id_i)=\frac{Frequency(id_i)}{Len}$$

$$P(T|id_i) \triangleq P(\tilde{T}|id_i) = \frac{Frequency(\tilde{T}|id_i)}{count_{id_i}}$$

$Frequency(id_i)$：id_i 在历史轨迹中出现的次数；

Len：历史轨迹长度；

$Frequency(\tilde{T}|id_i)$：经过卡口 id_i 的与轨迹 Track 相似的轨迹 $\tilde{T}$ 出现的次数；

相似度定义为：$sim = \frac{len[T \cap \tilde{T}]}{len[T]}$。

$count_{id_i}$：经过卡口 id_i 的次数。

2.4.3 算法实现

A. 贝叶斯算法

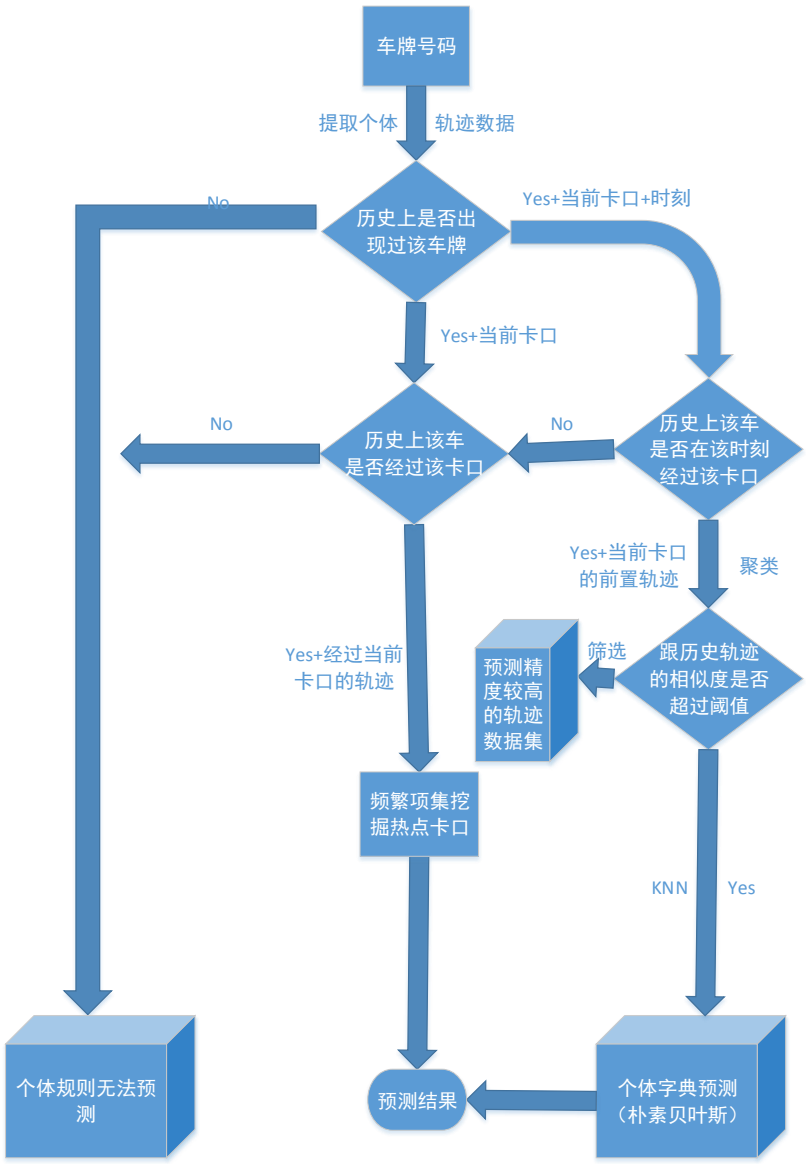

图 2 方案 A 模型框架

阶段一：构造训练集与测试集

Step1：取前六个月的轨迹作为训练集 data_train，后一个月的轨迹作为测试集 data_test，训练集与测试集的比例为 9:1。

（1）考虑到车辆在经过某个卡口之后，两个小时之内不会再被拍到的情况，对 data_train 做如下处理：轨迹末位补 0，作为不会再被拍到的情况，即

（车辆 id，车牌颜色）：（卡口 id，经过时间）；……；（0，经过时间（任意））

（2）无论车辆处于轨迹中的哪个卡口，其接下来的轨迹都是可以被预测的，因此，对

data_test 做如下处理：对经过 5 个（例）卡口的轨迹，进行切分（为了方便后续操作，统一轨迹长度为 6）

$$A\to B\to C\to D\to E\Rightarrow\begin{cases}0\to 0\to 0\to 0\to 0\to A\to (B,C,D)\\0\to 0\to 0\to 0\to A\to B\to (C,D,E)\\0\to 0\to 0\to A\to B\to C\to (D,E,0)\\0\to 0\to A\to B\to C\to D\to (E,0)\\0\to A\to B\to C\to D\to E\to (0)\end{cases}$$

阶段二：模型训练

Step2：根据 data_train 数据集中，车辆的个体历史轨迹建立模型，在车辆历史轨迹中按照“时间&卡口”进行聚类。

对于时间，可从时间戳中提取二维时间信息：

（1） 是否工作日（week），是取 0，否取 1；

（2） 将一天分为 6 个时间段，即 0-4 点、4-8 点，……，20-24 点，时间处于哪个时间段（time），取值为 0、1、2、3、4、5。

提取完时间信息，将该辆车在某一时刻（week，time）经过某一卡口（A）的所有历史轨迹聚在一起，得到该辆车在（(week，time)，A）之后可能去的所有卡口，以及去这些卡口的次数，并对这些卡口按照次数排序。保存结果 neighbor_time，提供轨迹预测的候选集 1，neighbor_time 形式如下：

car：{ ((week，time)，A)：{A1：count，A2：count，……}，((week，time)，B)：{B1：count，B2：count，……}，……}

Step3：类似 Step2，考虑车辆小部分违背时间规律的轨迹，舍弃时间信息，将车辆按照“卡口”聚类，得到 neighbor_no_time，提供轨迹预测的候选集 2，neighbor_no_time 形式如下：

car：{A：{A1：count，A2：count，……}，B：{B1：count，B2：count，……}，……}

阶段三：模型预测

Step4：新建数据集 1，针对 data_test 中的预测轨迹，判断车辆 id 在 data_train 中是否出现过，若没有出现过，则将此类轨迹归入数据集 1；若出现过，继续下面的步骤。

Step5：针对 Step5 筛选的可预测集，根据预测轨迹的车辆 id 在 data_train 中筛选出同一辆车的历史轨迹，提取车辆最后经过的卡口及时间（(week，time)，A），判断该辆车的历史轨迹中是否出现过（(week，time)，A），若出现过，进入 Step6；若没有出现过，进入 Step8。

Step6：应用 KNN 算法，在历史轨迹中寻找出与预测轨迹相似度大的轨迹，轨迹相似度定义如下：

$$Similarity = length（历史轨迹 \cap 预测轨迹）/length（预测轨迹）$$

设置相似度阈值 T，统计经过候选集 1 中的卡口且 Similarity >= T 的历史轨迹条数 count_sim_tra 并保存；

将候选集 1 中的每个卡口按照 count_sim_tra 排序，输出 count_sim_tra 最大的前三卡口（若候选集 1 不足三个，则全部输出），得到预测结果集 result。

下面举例说明：

对于预测轨迹 —（车辆 id，车牌颜色）:（卡口 A，t1）;（卡口 B，t2）;……;（卡口 C，t3）

需获得以下信息：车辆 id、车牌颜色、最后经过卡口信息（(week，time)，C）、轨迹卡口信息（A,B,……,C）。

（1） 根据车辆（车辆 id，车牌颜色）获取车辆历史轨迹，结合（(week，time)，C），根据 Step2 得到的 neighbor_time，获得候选集 1；

（2） 对于候选集 1 中所有卡口，分别统计经过这些卡口且与（A,B,……,C）的轨迹相似度 Similarity >= T 的历史轨迹条数 count_sim_tra；

（3） 对候选集 1 中所有卡口按 count_sim_tra 排序输出。

Step7：若 result 的卡口数量不足 3，则用轨迹预测候选集 2 补全；否则，用轨迹预测候选集 2 中次数最多的卡口替换掉 result 中概率最小的卡口（三个卡口不重复）。输出最终

预测结果 Result。

Step8：该步骤用到 neighbor_no_time，对于待预测轨迹，只提取其最后经过的卡口信息 A，判断该辆车的历史轨迹中是否出现过 A，若出现过，进入 Step9；若没有出现过，则将此类轨迹归入数据集 1，进入 Step10。

Step9：同 Step6。

Step10：对数据集 1，需根据群体轨迹构造新的候选集 3，即根据前 6 个月的车辆轨迹信息，统计车辆经过每个卡口后可能到达的所有卡口以及到达次数，得到 neighbor_all，取到达次数最多的前 10 个卡口，构成候选集 3。

Step11：若 C 在群体轨迹中出现过，则同 Step6 可推出三个预测卡口；否则，该轨迹不可预测。

最后输出的结果形式为：

（车辆 id，车牌颜色）：[(预测卡口 1，相对概率)，(预测卡口 2，相对概率)，(预测卡口 3，相对概率)]

B. 轻量级贝叶斯算法

在方案 A 中，需要将训练集轨迹进行切割以及末位补 0 的操作，以体现车辆经过某一卡口后，超过两小时未出现，然而在轨迹预测这一场景中，对轨迹进行切割的操作并不是必须的，若将车辆消失超过两小时看作车辆去了一个虚拟卡口 0，则可以通过插入虚拟卡口达到轨迹切割同样的效果。

在该方案中，数据预处理方法改变如下：

若连续经过的两个卡口的时间间隔大于 2h，则两个卡口之间插入虚拟卡口 0，如

$$car:(8:00am,A),(9:00am,B),(13:00pm,C),(14:00pm,D)$$
$$\rightarrow car:(8:00am,A),(9:00am,B),(9:00am,0),(13:00pm,C),(14:00pm,D)$$

阶段一：构造训练集与测试集

Step1:取前六个月的轨迹作为训练集 data_train,后一个月的轨迹作为测试集 data_test，

训练集与测试集的比例为 9:1。

（1） 构造训练集:

将车辆前六个月的轨迹拉成一条，根据时间间隔插入虚拟卡口，按照“车牌号+车牌颜色”建立索引，生成数据集如下，

（车辆 id，车牌颜色）:（卡口 id，经过时间）;（0，经过时间）;（卡口 id，经过时间）;……

（2） 构造测试集:

根据方案 A 的经验，取轨迹长度为 3 时，可达到算法精度与效率的最高，即

$$A\to B\to C\to D\to E\Rightarrow\begin{cases}0\to 0\to A\to (B,C,D)\\ 0\to A\to B\to (C,D,E)\\ A\to B\to C\to (D,E,0)\\ B\to C\to D\to (E,0)\\ C\to D\to E\to (0)\end{cases}$$

阶段二：模型训练

在初始版本中，分开计算 $P(id_i)$ 与 $P(T|id_i)$，因此需要首先构造候选集，根据候选集计算 $P(id_i|T)$，对概率进行排序输出 topN。然而，在轨迹预测中，我们可以发现

$$\begin{aligned}P(id_i|T)&=P(id_i)P(T|id_i)\\&=\frac{Frequency(id_i)}{Len}\cdot\frac{Frequency(\tilde{T}|id_i)}{count_{id_i}}\\&=\frac{Frequency(\tilde{T}|id_i)}{Len}\end{aligned}$$

对于给定车辆查找到的历史记录， Len 是固定的，因此只需确定 $Frequency(\tilde{T}|id_i)$ 。

Step2：对于 data_test 数据集中每一条待预测轨迹(长度为 3)：

（车辆 id，车牌颜色）:（卡口 id_1，经过时间）;（卡口 id_2，经过时间）;（卡口 id_3，经过时间）

提取如下三个特征：

1、（车辆 id，车牌颜色）；

2、待预测轨迹 T:[卡口 id_1，卡口 id_2，卡口 id_3]；

3、经过的最后一个卡口及其经过时间：（last_id，t）.

对于时间，可从时间戳中提取二维时间信息：

（1） 是否工作日（week），是取 0，否取 1；

（2） 处于一天中的哪一个小时。

Step3：根据 data_test 第一个特征（车辆 id，车牌颜色），从 data_train 中筛选其历史轨迹,如

（车辆 id，车牌颜色）：（A，经过时间 t1）；（0，经过时间 t2）；（B，经过时间 t3）；（C，经过时间 t4）；（D，经过时间 t5）；（E，经过时间 t6）；（F，经过时间 t7）

进行特征分离：

1、[A,0,B,C,D,E,F]；

2、[t1,t2,t3,t4,t5,t6,t7].

Step4：筛选出历史轨迹之后，利用待预测轨迹在历史轨迹上作滑窗遍历，示意图如下：

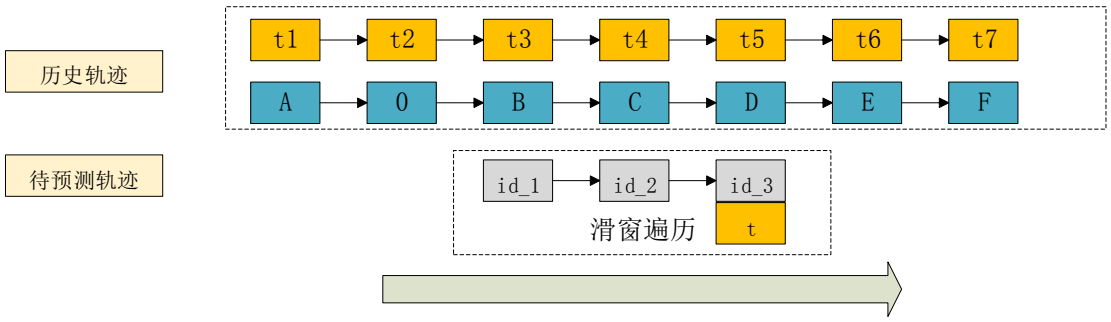

图 3 方案 B 模型预测过程

遍历直到待预测轨迹最后一个卡口 last_id 等于历史轨迹中的某一个卡口，以上图为例，若 id_3 = D，则取出 D 的下一个卡口 E 作为候选集之一id_i，取 D 之前三个卡口连同 D 组

成相似轨迹$\tilde{T}$，同时计算该位置这一候选集的后验概率$P(id_i|T)$，即计算$Frequency(\tilde{T}|id_i)$。

对于遍历过程中遇到的所有id_i，根据轨迹相似度，为区分不同相似程度的轨迹对候选集的贡献，给出如下定义：

$$Frequency(\tilde{T}|id_i)=\sum_{j=1}^{Frequency(id_i)} f(\tilde{T}|id_{ij})$$

$$f(\tilde{T}|id_{ij})=\begin{cases}\alpha_1\cdot\beta, & iflen[T\cap\tilde{T}]=1\\ \alpha_2\cdot\beta, & iflen[T\cap\tilde{T}]=2\\ \alpha_3\cdot\beta, & iflen[T\cap\tilde{T}]=3\end{cases}$$

其中，$\alpha_1<\alpha_2<\alpha_3$，$\beta$的取值取决于待预测轨迹最后一个卡口的时间（week＋hour），与历史轨迹重合卡口时间（week＋hour）的关系。

阶段三：模型预测

Step5：根据$Frequency(\tilde{T}|id_i)$进行排序，选取 top3 作为预测结果。

Step6：生成候补候选集：遍历所有训练集（即过去六个月所有车辆的过车轨迹），找出每个卡口之后最可能出现的三个卡口。改候选集主要用于应对以下几种情况：

1、待预测车辆在过去六个月从未出现过；

2、待预测轨迹最后一个卡口，在该车辆的历史轨迹中从未出现过；

3、根据$Frequency(\tilde{T}|id_i)$选出的候选集不足 3 个。

最后输出的结果形式为：

（车辆 id，车牌颜色）：[(预测卡口 1，相对概率)，(预测卡口 2，相对概率)，(预测卡口 3，相对概率)]

2.5 算法评估

评估指标一，将虚拟卡口 0 看做正常卡口，最终预测结果给出的三个卡口，与预测车辆后

来实际去过的三个卡口，有一个重合，即认为预测正确；

如：对于预测轨迹 — 车辆 id：（卡口 A，t1）；（卡口 B，t2）；……；（卡口 C，t3）

在（卡口 C，t3）之后，预测卡口序列 l = [C1,C2,C3]，实际卡口序列 L = [C4,C5,C6]。

若 length（l∩L）= 0，则预测错误；否则预测正确。

评估指标二，将虚拟卡口 0，即接下来 2h 车辆消失，看做一种特殊情形，同样有 l 与 L，分为以下几种情况：

1、预测为消失，实际真的消失，预测正确：C1 = C4 = 0；

2、预测为消失，实际往后开了一个卡口或更多，预测错误：C1 = 0 != C4；

3、预测为继续前行，给出三个预测卡口，实际往后开，且给出的卡口预测正确，预测正确：C1 != 0 != C4 且 length（l∩L）!= 0；

4、预测为继续前行，实际消失，预测错误：C1 != 0 = C4；

5、预测为继续前行，给出三个预测卡口，实际确实往后开，但给出的卡口预测错误，预测错误：C1 != 0 != C4 且 length（l∩L）= 0。

3. 总结

对于交通轨迹预测这种实时性的业务需求，我们在关注算法精确度的同时，更应该关注算法的复杂度，因此特征的选择及应用，数据的处理，算法的训练预测过程，都应该做到简介有效。同样是基于朴素贝叶斯的算法方案，方案 B 在方案 A 的基础上，大大提高了算法效率，同时保证了算法的精度。

4. 修订记录

序号	变更时间	版本	变更人	审批人	变更说明
1	2019-2-28	V 0.1.0	辛杰		新建

附件

附件 1. 车牌号码相关规则

规则	备注
长度必须为 7 或者 8（下述规则都是针对 7 个字符的	车牌号码为 7 个或 8 个字符（新能源车等）。
第一个必须是汉字，且必须是“备注”中 31 个汉字中的某一个	京、鄂、津、湘、冀、粤、晋、桂、蒙、琼、辽、渝、吉、川、黑、贵、沪、云、苏、藏、浙、陕、皖、甘、闽、青、赣、宁、鲁、新、豫
结尾如果不是字母或者数字，必须是“备注”中9 个汉字的某一个	领、使、警、学、挂、港、澳、试、超
不能含有字母“I”和“O”	车牌号码不含有这两个字母
字母和数值的组合只能是“备注”中的3种组合	全部为数字
	只有 1 个字母，其余为数字
	如果有两个字母，字母只能出现在如下两个位置上（3、4），（3、5），（3、7），（4、5），（6、7）
号牌中不能含有除了规定字符以外的其他字符	规定字符包括上述备注中允许的40 个汉字，0-9 共计10 个数字，以及A-Z 累计24 个字母（I 和O 除外）