

视频目标轨迹分析的改进 PrefixSpan 方法

袁和金

YUAN Hejin

华北电力大学 计算机系, 河北 保定 071003

Department of Computer, North China Electric Power University, Baoding, Hebei 071003, China

YUAN Hejin. Modified PrefixSpan algorithm for video target trajectory analysis. *Computer Engineering and Applications*, 2011, 47(32): 7-10.

Abstract: A modified PrefixSpan algorithm is put forward to analyze the video target's movement through their trajectories in this paper. In the algorithm, the trajectories are recoded through vector quantization. The modified PrefixSpan algorithm is utilized to mine the frequent and continuous patterns from them. An approximate string matching method is given to detect whether the video target's movement is abnormal or not. Since this method needn't to measure similarity among the trajectories, it can be well used in occasion where the trajectory set scale is very large or the num of trajectory distribution pattern is difficult to determine. The experiments on the trajectories of different scenes show that the method is effective.

Key words: trajectory pattern; sequential pattern mining; PrefixSpan algorithm

摘 要: 从序列模式挖掘的角度对视频目标运动轨迹的分析和应用问题进行了研究, 提出了一种基于改进 PrefixSpan 的频繁轨迹模式挖掘算法, 并给出了基于所挖掘的频繁模式进行在线目标运动异常检测的方法。该方法对目标的运动轨迹进行量化编码, 采用改进的 PrefixSpan 算法挖掘其中连续出现的频繁模式, 通过字符串近似匹配的方法来检测当前运动轨迹所表示的目标行为是否异常。由于不需要计算两两轨迹之间的相似性, 该方法可以应用于规模较大、分布模式数目难以确定场合下的视频目标轨迹分析问题。对仿真和真实场景的实验验证了该方法的有效性。

关键词: 轨迹模式; 序列模式挖掘; PrefixSpan 算法

DOI: 10.3778/j.issn.1002-8331.2011.32.003 文章编号: 1002-8331(2011)32-0007-04 文献标识码: A 中图分类号: TP393

1 引言

运动轨迹的分布模式是视频目标行为特征的重要表现方式之一, 在许多视觉监控场合中通过对目标运动轨迹的分析就可以提取出目标的行为模式。近年来, 基于轨迹的目标行为分析方法引起了国内外学者的广泛关注, 并已提出了许多不同的方法, 如基于轨迹之间不同距离度量方式的聚类分析方法^[1]、基于轨迹特征表示的聚类分析方法^[2]和基于轨迹分布建模的聚类分析方法^[3-4]等。这些方法大都需要度量不同轨迹之间的相似性并据此进行聚类。然而, 不同目标的运动轨迹在长度和运动速度等方面都有差别, 很难以一种非常自然的方式来度量它们之间的相似性。另外, 当场景中目标的运动比较复杂时, 很难对它们进行准确的聚类, 聚类数目的确定也比较困难。再者, 目标在场景中的运动既表现出很大的规律性, 同时也有一定的随机性。这种随机性的运动也会影响到基于聚类的轨迹分析方法的效果。

其实, 基于轨迹的视频目标行为分析的关键在于如何从给定的轨迹中提取出目标在场景中运动的典型模式, 并能基于这些模式对未知的目标运动进行异常检测即可, 从本质上讲它并不需要通过聚类将其准确地划分为不同类别这一复杂

的过程。因此, 可以从序列模式挖掘的角度出发, 寻找和发现给定的轨迹集合中频繁出现的轨迹序列或片段, 并将其作为目标在场景中运动的典型规律, 用于检测场景中目标的异常行为。然而, 这方面的工作还没有引起计算机视觉领域学者们的足够重视, 目前报道的文献还不多。代科学^[5]等提出了一种用于发现视频目标频繁轨迹模式的挖掘算法。该方法通过高曲率的特征点来表示目标的运动轨迹并将轨迹模式的支持度定义为与它等长的所有轨迹段的相似度和, 然后采用 Apriori 算法挖掘二维轨迹数据库中的频繁模式。然而, 文中支持度的定义与一般序列模式挖掘中所采用的序列出现频率明显不同。另外, 作者也没有讨论所挖掘出的频繁轨迹模式的应用问题。Yang^[6]和 Morzy^[7]等针对移动计算环境下运动目标的位置预测问题, 提出了一种从历史轨迹数据中挖掘目标运动规律的方法, 即首先将运动区域划分成面积相等的若干个方格, 然后将目标运动轨迹离散化成由经过这些方格的边所组成的有序序列, 接着采用 GSP 算法来挖掘其中的频繁序列模式并生成推理规则, 最后基于不同策略对目标未来的运动趋势进行预测。Gianotti^[8]详细探讨了由 GPS 设备获取的运动轨迹中频繁序列模式的挖掘问题, 给出了关于轨迹模式挖

基金项目: 中央高校基本科研业务费专项资金资助(No.10QG21)。

作者简介: 袁和金(1977—), 男, 博士, 副教授, 主要研究领域为模式识别和计算机视觉。E-mail: yhj_1977@163.com

收稿日期: 2011-05-26; 修回日期: 2011-07-26

掘的一些形式化定义和算法实例。与Yang等人的工作类似,其频繁轨迹模式中除了考虑目标依次经过各区域的顺序之外,更关心的是目标在各个区域的逗留时间。虽然,以上方法分别从不同的角度考虑了轨迹数据库中频繁模式的挖掘问题,然而由于它们并不是专门针对视觉监控问题而设计的,也没有考虑基于轨迹模式的目标异常运动检测等问题,因而并不能完全满足视觉监控中运动轨迹分析和应用的要求。

本文从序列模式挖掘的角度对视频目标的轨迹分析问题进行了研究,提出了一种基于改进PrefixSpan的轨迹频繁模式挖掘算法,并给出了基于挖掘出的频繁轨迹模式进行在线检测目标异常运动的方法。

2 基于改进PrefixSpan的频繁轨迹模式挖掘算法

由于轨迹反映的是目标在监控场景中连续运动时的规律,因此在轨迹序列模式的挖掘中,只有如图1(a)所示的连续的轨迹片段才能作为频繁序列模式被挖掘出来,而图1(b)所示的非连续片段则是不允许的。正是这一显著差异使得现有的序列模式挖掘算法并不能直接应用到轨迹序列模式挖掘中,需要进行相应的修改。鉴于Han^[9]等人提出的PrefixSpan(Prefix projected Sequential pattern mining)算法是目前序列模式挖掘中性能较好的一种方法,本文对其进行了改进,使其能够用于连续的轨迹序列模式挖掘。基于改进PrefixSpan的轨迹模式挖掘算法首先通过流矢量化编码将轨迹变换成符号序列的集合,然后采用PrefixSpan投影挖掘的思想找出其中频繁出现的序列模式作为轨迹分布的表示。

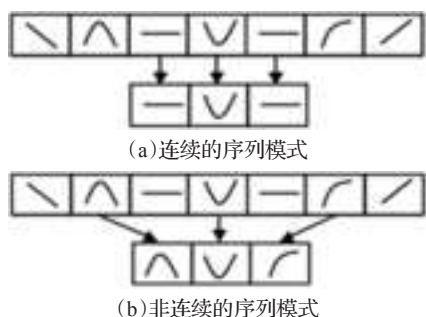


图1 序列模式示意图

2.1 轨迹流矢量的符号化

目标的运动轨迹 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ 是由不同时刻目标质心坐标组成的有序序列,其中还隐含着目标的运动速度、加速度等信息,而现有的序列模式挖掘算法大都是对符号序列进行挖掘的,因此不能直接采用上述方法进行处理,需要对其进行符号化。本文采用模糊K均值方法对轨迹中的所有流矢量 $f_i = (x_i, y_i, dx_i, dy_i)$ (轨迹在第*i*个时刻的坐标和速度)进行聚类,然后将每个流矢量用距离其最近的聚类中心的序号来编码。这样,就可以把由多维实数序列表示的运动轨迹变换成简单的符号序列。与标准序列模式挖掘算法不同的是,这里没有项集,每个元素都是由单个的流矢量化后的符号组成,这将在很大程度上简化轨迹序列模式挖掘算法的实现。另外,在序列模式挖掘中,要求频繁模式必须完整、准确地包含在轨迹序列中,而实际跟踪到的不同目标的运动轨迹之间总会存在一些差异。因此,为了避免挖掘出的频繁模式仅是一些很短的轨迹片段,本文将编码后轨迹序列中连续出

现多次的符号仅表示1次。这样不仅可以减少挖掘算法的运行时间,而且能够使得目标在场景中运动的频繁轨迹模式都能被有效地挖掘出来。

2.2 改进的PrefixSpan挖掘算法

文献[9]提出了基于前缀投影的PrefixSpan算法,它是一种不需要产生候选频集的频繁序列模式挖掘算法,其基本思想是:先扫描序列数据库找出各频繁项,产生投影数据库的集合,其中每个投影数据库关联一个频繁项,然后重复上述过程对每个投影库进行单独挖掘。由于每次只需要扫描投影数据库,所以PrefixSpan算法具有很高的效率。下面结合PrefixSpan,给出一些与轨迹序列模式挖掘相关的基本概念。

定义1 对序列 $\alpha = \langle a_1, a_2, \dots, a_p \rangle$ 和 $\beta = \langle \beta_1, \beta_2, \dots, \beta_q \rangle, p \leq q$, 如果存在某个递增的下标序列 $i_1, i_2, \dots, i_p (i_1 < i_2 < \dots < i_p)$, 使得 $a_1 = \beta_{i_1}, a_2 = \beta_{i_2}, \dots, a_p = \beta_{i_p}$, 则称序列模式 α 是 β 的子序列,或 β 包含 α 。

定义2 给定序列数据库 SD , 若一序列 $s \in SD$ 包含序列模式 α , 则称序列 s 支持序列模式 α 。

定义3 给定序列数据库 SD 和序列模式 α , SD 中支持 α 的序列数目称为 α 的支持数,记为 α_{Count} 。 α_{Count} 与 SD 中序列的总数之比称为序列模式 α 的支持度,记为 α_{Sup} , 即

$$\alpha_{\text{Sup}} = \frac{\alpha_{\text{Count}}}{\|SD\|}$$

定义4 给定序列数据库 SD 和序列模式 α , 如果 $\alpha_{\text{Sup}} \geq \text{minsupp}$, 则称 α 为频繁序列模式,其中 minsupp 为用户指定的最小支持度。

以上是一般序列模式挖掘中常用的关于频繁序列模式及其支持度的定义。然而不同于交易数据库中顾客购买行为模式等的挖掘,如图1(a)所示,轨迹分布模式应该是连续的序列片断,因此,不能直接应用PrefixSpan算法对轨迹序列进行挖掘,而应该对其进行限定,只挖掘那些连续的频繁序列模式。为此,需要对定义1中的“包含”关系进行如下的限制。

定义5 对于序列 $\alpha = \langle a_1, a_2, \dots, a_p \rangle, \beta = \langle \beta_1, \beta_2, \dots, \beta_q \rangle, p \leq q$, 如果存在某个整数 i , 使得 $a_1 = \beta_i, a_2 = \beta_{i+1}, \dots, a_p = \beta_{i+p-1}$, 则称序列模式 α 是 β 的子序列,或称 β 包含 α 。后续的论述中,关于轨迹序列的“包含”都采用该定义。

在上述“包含”关系的基础上,定义与之对应的“前缀”、“后缀”和“投影”等概念如下。

定义6 给定两个轨迹序列模式 $\alpha = \langle a_1, a_2, \dots, a_p \rangle$ 和 $\beta = \langle \beta_1, \beta_2, \dots, \beta_q \rangle, p \leq q$, α 称为 β 的前缀当且仅当 $a_1 = \beta_1, a_2 = \beta_2, \dots, a_p = \beta_p$ 。

定义7 给定轨迹序列模式 α 和 $\beta (\beta \subseteq \alpha)$, α' ($\alpha' \subseteq \alpha$) 称为 α 对应于 β 的投影,当且仅当同时满足以下两个条件:

- (1) β 是 α' 的前缀。
- (2) α' 是 α 满足条件(1)的最大子序列。

定义8 设 $\alpha' = \langle a_1, a_2, \dots, a_n \rangle$ 为 α 对应于 $\beta = \langle \beta_1, \beta_2, \dots, \beta_m \rangle$ 的投影,则序列 $\langle a_{m+1}, a_{m+2}, \dots, a_n \rangle$ 为 α 对应于序列 β 的后缀(根据投影和前缀的定义,显然有 $a_1 = \beta_1, a_2 = \beta_2, \dots, a_m = \beta_m$)。

定义9 设 α 为一轨迹序列模式,则 α -投影数据库为 SD 中以 α 为前缀的轨迹序列对应于 α 的后缀组成的集合,记为 $SD|_{\alpha}$ 。

由定义9可以看出,在计算序列模式 α 的投影数据库时,只有以 α 为前缀的那些序列才能包含在其投影数据库中,即

只有当待投影序列的第一个符号和前缀的最后一个符号相同时才进行投影操作。这样就能保证挖掘出的频繁轨迹模式都是连续的轨迹片段,这也是本文对PrefixSpan算法进行改进以使其能够应用于轨迹序列模式挖掘的关键所在。

基于上述定义,本文提出了包含如下4个阶段的改进PrefixSpan轨迹频繁模式挖掘算法:(1)首先挖掘所有的1-频繁序列模式;(2)按照挖掘出的1-频繁序列模式对轨迹数据库进行重组(对给定的轨迹序列 s 和1-频繁序列模式 α ,将 s 中从 α 第一次出现的位置到 s 末尾的子串加入到 α 对应的重组子集),将轨迹序列模式挖掘的任务分解到重组后的不同轨迹子集上;(3)对重组后的各个轨迹子集,按定义5中关于轨迹序列“包含”关系的定义,通过不断的投影和模式生长挖掘其中的频繁序列模式;(4)对挖掘出的频繁序列模式进行筛选,找出其中的最大频繁序列模式作为最终的挖掘结果。

下面通过一个具体的例子来说明采用修改后的方法与标准PrefixSpan算法之间的差别。设有经符号化处理后的轨迹序列集合 $SD=\{<1,2,3,6,7>,<2,3,6,7>,<4,5,3,8,9>,<1,2,3,6>,<4,5,3,9>,<5,3,8,9>\}$,其中 $I=\{1,2,3,4,5,6,7,8,9\}$ 是项的集合,最小支持度 $minsupp=2/6=33.33\%$,则基于改进PrefixSpan算法的轨迹序列模式挖掘过程如下:

(1)扫描轨迹序列数据库一次,得到频繁1-频繁序列模式及其支持数。

(2)根据 f_list 的结果对轨迹序列数据库进行重组,得到9个子序列集合。

(3)根据第(2)步的处理结果将序列模式挖掘的任务分解为9个相互独立的子任务,即从中分别挖掘前缀为 $<1>,<2>,\dots,<9>$ 的频繁序列模式。

(4)扫描 S_1 一次,得到前缀为 $<1>$ 的频繁2-序列模式,结果为 $<1,2>$,然后据此构造 $<1,2>$ 的投影数据库为 $SD|_{<1,2>}=\{<3,6,7>,<3,6>\}$;对它继续挖掘可以得到频繁序列模式 $<1,2,3,6>$,此时投影数据库为 $SD|_{<1,2,3,6>}=\{<7>\}$ 。其中序列的个数已经小于 $\|SD\| \times minsupp=2$,不可能再生成长出更长的频繁序列了,对 S_1 的挖掘结束。

(5)对 S_2, S_3, \dots, S_9 采用与第(4)步类似的处理,分别得到前缀为 $<2>,<3>,\dots,<9>$ 的频繁序列模式,结果如表1所示。

(6)对挖掘出的频繁序列模式集进行删减,从中挑选出所有的最大频繁序列模式作为最终的挖掘结果。

对该序列数据库采用标准的PrefixSpan算法进行挖掘,结果如表2所示。其中,表格中列出的频繁序列模式均为各投影集下的最大频繁序列模式。显然,其中包含了一些非连续出现的频繁序列模式,如表中的“☆”标示之处。

表1 改进后算法的挖掘结果

前缀	频繁序列模式
$<3>$	$\{3,6,7\},<3,8,9>$
$<2>$	$<2,3,6,7>$
$<5>$	$<5,3,8,9>$
$<6>$	$<6,7>$
$<9>$	$<9>$
$<1>$	$<1,2,3,6>$
$<4>$	$<4,5,3>$
$<7>$	$<7>$
$<8>$	$<8,9>$

表2 标准PrefixSpan算法的挖掘结果

前缀	频繁序列模式
$<3>$	$\{<3,7>,<3,9>,<3,6,7>,<3,8,9>\} \star$
$<2>$	$\{<2,7>,<2,3,7>,<2,3,6,7>\} \star$
$<5>$	$\{<5,3,8,9>\} \{<5,3,9>\} \star$
$<6>$	$\{<6,7>\}$
$<9>$	$\{<9>\}$
$<1>$	$\{<1,6>,<1,2,6>,<1,2,3,6>\} \star$
$<4>$	$\{<4,9>,<4,5,9>,<4,5,3,9>\} \star$
$<7>$	$\{<7>\}$

3 基于字符串近似匹配的在线异常检测

视觉监控系统的一个重要目标是能够自动识别出监控场景中出现的非正常或非预期的目标运动或行为,而且这种识别应该能够在异常行为将要或正在发生时就可以完成。提取出目标在场景中运动的频繁轨迹模式之后,就可以基于它们来判断当前目标的运动是否异常。具体而言,就是将当前正在跟踪目标的运动轨迹与挖掘出的各频繁轨迹模式进行匹配,如果与各频繁模式的偏差都比较大,则认为当前目标的运动属于异常。这种检测方法不依赖于特定的监控场景,通用性较强。上述异常检测过程的关键在于如何将符号化后的测试轨迹与挖掘出的频繁模式进行匹配。由于量化编码化后的测试轨迹和挖掘出的频繁模式均为字符串,因此可以采用简单的字符串匹配算法。但由于目标在场景中出现的位置、持续的时间以及运动速度差异等因素的影响,待测试的运动轨迹可能无法与任何一个频繁序列模式精确匹配。因此,应该采用一种具有较强容错能力的近似匹配算法。Sun和Manber^[10]给出了一种非常灵活的字符串近似匹配的快速算法,该算法在进行字符串匹配时允许包含一定数量的插入、删除和替换误差。因此,本文采用该算法来对待测试轨迹和频繁轨迹模式进行近似匹配。

设 $\Sigma=\{c_1, c_2, \dots, c_{|\Sigma|}\}$ 为字母表,对应本文中不同的流矢量符号编码; P 为待匹配的序列模式,对应本文中的待测试轨迹序列,长度为 m ; T 为参与匹配的文本字符串,对应本文中的某个频繁轨迹模式,长度为 $n(n \geq m)$ 。 R 是一长度为 m 的位向量, R_j 表示当前搜索位置 j 处 P 和 T 的匹配情况。若 $R_j[i]=1$,则表示由 P 的前 i 个子符组成的序列和 T 中从第 $j-i+1$ 到 j 之间的字符串能够匹配,即 $p_1 p_2 \dots p_i = t_{j-i+1} t_{j-i+2} \dots t_j$ 。如果 $R_j[m]=1$,则 P 和 T 完全匹配(即 P 包含在 T 中)。

当匹配过程中允许 P 和 T 之间有 d 个插入、删除或替换误差时,近似匹配方法如下:

设 $R_0^d=11\dots100\dots0$ (其中包含 d 个1和 $m-d$ 个0),则根据 R_j^d 计算 R_{j+1}^d 的过程如下式所示:

$$R_{j+1}^d = Rshift(R_j^d) \wedge S_c \vee Rshift(R_j^{d-1}) \vee Rshift(R_{j+1}^{d-1}) \vee R_j^{d-1} = Rshift(R_j^d) \wedge S_c \vee Rshift(R_j^{d-1} \vee R_{j+1}^{d-1}) \vee R_j^{d-1}$$

其中, S_c 表示 P 中第 i 个位置上的字符 c 对应的位向量, R_j^d 表示在允许 d 个误差的情况下, $p_1 p_2 \dots p_i$ 和 $t_{j-i+1} t_{j-i+2} \dots t_j$ 之间的匹配情况。

4 仿真结果及分析

为了测试本文方法的有效性,进行了两组实验。

实验1 频繁轨迹模式的挖掘实验

用于实验的运动轨迹如图2(a)和(c)所示,其中场景1为仿真场景,过手工的方法生成了一些运动轨迹用于频繁模式挖掘和异常检测;场景2中的轨迹为文献[1]中的实验数据。图中蓝色的曲线为目标的目标运动轨迹,绿色圆点表示轨迹的起点,黄色圆点表示轨迹的终点。图2(b)和(d)给出了对上述两处场景中的轨迹流矢量进行量化编码时,量化中心(X 和 Y 坐标)的示意图。对各场景中的运动轨迹分别采用标准PrefixSpan算法(简称算法I)和本文改进后的算法(简称算法II)进行挖掘后,得到的频繁轨迹模式如图3(a)~(d)中的箭头连线所示,其

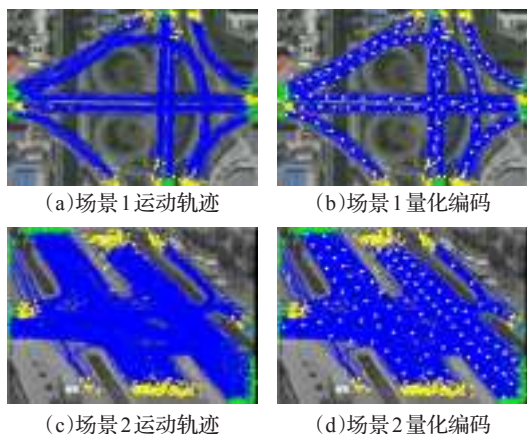


图2 不同场景中的运动轨迹

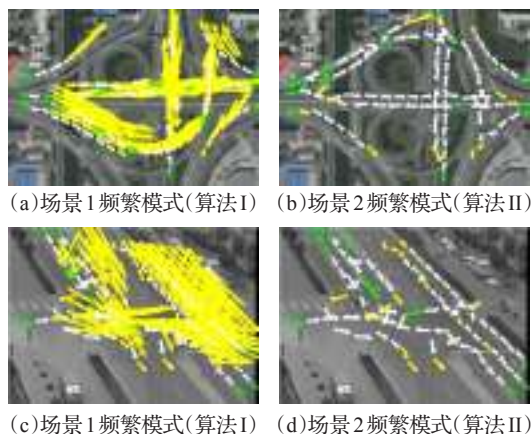


图3 轨迹序列模式的挖掘结果

中绿色的箭头表示频繁轨迹模式的起点,黄色的箭头为终点。显然,本文给出的挖掘算法能够比较准确地发现各场景中目标的运动规律,而且运行时间较少,尤其是当轨迹的数目比较大的时候,这种优势更为明显;而标准的PrefixSpan算法由于没有考虑轨迹模式的连续性要求,挖掘出了很多间断的轨迹片断。

实验2 基于频繁轨迹模式的目标异常运动检测实验

从前面对异常检测算法的介绍可知,其性能不仅与待测试的部分轨迹的长度有关,而且也与检测过程中所允许的匹配误差有关。由于受交通规则、目标实际运动和实验条件等因素的限制,很难在实际场景中获取足够多的异常轨迹来对算法的性能进行定量分析。因此,这里通过手工的方法生成了一些仿真场景的测试数据,用于异常检测实验,并采用如下的两个指标来进行评价。

检测率(True Positive Rate, TPR):检测出的真实异常轨迹的数目与异常轨迹总数的比值。

误报率(False Positive Rate, FPR):被误报为异常的正常轨迹数目与正常轨迹总数的比值。

在待测试轨迹均为完整轨迹的条件下,异常检测算法的TPR与FPR和检测过程中所允许的匹配误差百分比之间的关系如图4所示。从图中可以看出,在30%的匹配误差条件下,检测正确率TPR可以达到90%以上,而误报率FPR则控制在10%的范围内。

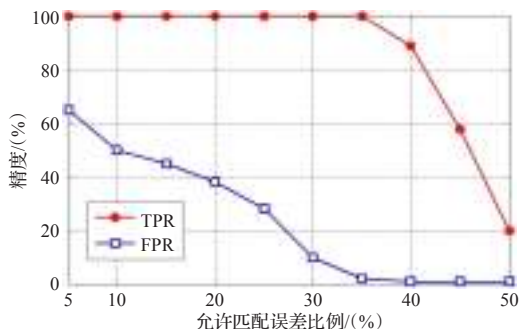


图4 异常检测算法的性能与允许的匹配误差之间的关系

为了进一步测试异常检测算法的在线应用效果,这里分别将测试轨迹按其长度的10%,20%,...,100%进行了实验,得到的TPR和FPR随测试轨迹长度百分比变化的关系如图5所示,其中所允许的匹配误差百分比设定为30%。从图中可以看出,在待检测轨迹长度百分比从10%到100%的增长过程

中,检测率迅速提高。当测试轨迹的长度百分比达到30%时,算法的检测率就可达到90%左右,而误报率则一直保持在很低的范围内。

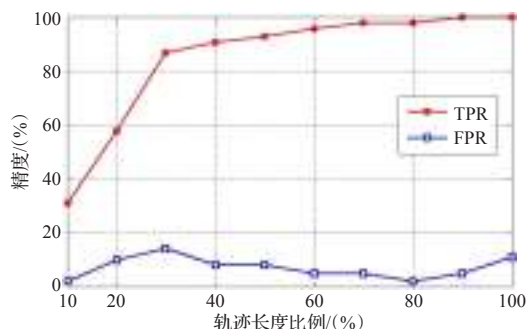


图5 异常检测算法的性能与轨迹长度之间的关系

5 结束语

本文将符号序列模式挖掘的思想引入到了轨迹分析中,并针对视觉监控中频繁轨迹模式挖掘的特殊要求提出了一种改进的PrefixSpan挖掘算法,然后给出了基于所挖掘的频繁轨迹模式进行目标异常行为检测的方法。相对于基于聚类的轨迹模式提取方法而言,该方法发现的模式较多且不同的模式之间存在着一些相同的片段。但该方法具有以下特点:(1)它不需要计算和存储两两轨迹之间的相似度,也不需要事先指定轨迹分布模式的数目,因而可以较好地应用于规模较大、分布模式数目不易确定场合下的轨迹分析问题;(2)通过轨迹序列和频繁模式之间的近似匹配,该方法在待测试轨迹较短的情况下也能将其中的异常检测出来,并把误报率控制在一个较低的范围内,具有很好的在线异常检测性能。

参考文献:

- [1] Hu W, Xiao X, Fu Z, et al. A system for learning statistical motion patterns[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(9): 1450-1464.
- [2] Bashir F I, Khokhar A, Schonfeld D. Real-time motion trajectory-based indexing and retrieval of video sequences[J]. IEEE Transactions on Multimedia, 2007, 9(1): 58-65.
- [3] 袁和金, 张艳宁, 周涛, 等. 基于归一化编辑距离和谱聚类的轨迹模式学习方法[J]. 计算机辅助设计与图形学学报, 2008, 6(2): 753-758.

(下转58页)