

基于漂移检测的流程变体差异分析方法

李 婷, 巩秀钢, 徐兴荣, 李会玲, 牛慧敏, 刘 聪

(山东理工大学 计算机科学与技术学院, 山东 淄博 255000)

摘 要:传统的流程挖掘技术通常假设过程模型在服务期间不会随时间改变,但是实际过程模型在执行中总是动态调整的。检测漂移发生的时间点并确定漂移过程中所涉及的行为变化,可以帮助业务分析人员更好地理解流程行为。目前大多数方法侧重于漂移检测和变体差异分析这两个独立的任务,但漂移检测仅能单纯定位发生变化的关键点,不能对漂移点前后流程行为的变化进行描述,因而无法有效地对流程进行配置和维护。因此,提出一种集漂移检测和流程变体差异分析于一体的综合方法,既能检测漂移又能对漂移后流程行为的差异进行具体分析。应用不同类型的控制流变化得到的事件日志来评估该方法,验证了所提方法的有效性。

关键词:漂移检测;过程模型;流程行为;变体差异分析;事件日志

中图分类号:TP391

文献标志码:A

Process variant analysis method based on drift detection

LI Ting, GONG Xiugang, XU Xingrong, LI Huiling, NIU Huimin, LIU Cong

(College of Computer Science and Engineering, Shandong University of Technology, Zibo, Shandong 255000, China)

Abstract: Traditional process mining techniques usually assume that the process model does not change over time during service, but in practice the process model is always dynamically adjusted during execution. Detecting the time point at which drift occurs and identifying changes in behavior involved in the drift process can help business analysts better understand process behavior. Most of the existing methods focus on the two separate tasks of drift detection and variant difference analysis. Drift detection can only locate the key points of change, but cannot describe the changes in process behavior before and after drift points in detail, so it cannot effectively configure and maintain the process. Therefore, a comprehensive method that integrates drift detection and process variant difference analysis is proposed, which can not only detect the drift but also analyze the difference of process behavior after the drift. The effectiveness of the proposed method is verified by applying the event log obtained from different types of control flow changes to evaluate the method.

Key words: drift detection; process model; process behavior; variant difference analysis; event log

过程挖掘技术提供了一种从事件日志中自动抽取过程知识的方法,其典型的应用场景之一是过程发现^[1],即从事件日志中挖掘出潜在的过程模型。过程发现算法能够从稳定行为的事件日志中生成过程模型,但由于多方面的因素,过程模型在真实的业务环境中是动态调整的^[2],这种业务流程在执行过程中的动态变化称为漂移。如果将传统的过程发现技术应用于含有漂移的事件日志中,挖掘出来的过程模型由于混杂了多个模型的行为变得复杂且无法解释。如何通过漂移检测来定位漂移前后流程内容的变化显得尤为重要。

收稿日期:2020-12-30

基金项目:国家自然科学基金项目(61902222);山东省泰山学者工程专项基金项目(tsqn201909109);山东理工大学引进高层次人才启动项目

作者简介:李 婷(1996—),女,山东济宁人,硕士研究生,主要从事过程挖掘、概念漂移等研究。

刘 聪(1990—),男,山东淄博人,教授,博士,主要从事过程挖掘、业务过程管理等研究,本文通信作者。

E-mail:liucongchina@163.com

在实践中,业务流程中漂移的存在是事先不知道的,分析人员需要随着时间的推移区分哪些流程发生了变化,哪些没有变化,从而做出决策来减轻其影响^[3]。现有的过程挖掘研究仅侧重于漂移检测,对漂移前后流程变化的分析和解释略显不足。本研究通过抽取日志中轨迹的活动关系来检测和定位漂移,对漂移前后的流程进行可视化分析,揭示具有显著差异的流程行为,帮助企业管理者适应或改进其业务流程。实验结果表明,所提方法能够有效检测到漂移,并对检测到的漂移提供准确描述。

1 相关工作

1.1 漂移检测

业务流程中事件日志的可用性激发了各种流程挖掘技术,这些技术主要支持过程监控和分析。经典的过程挖掘技术通常假设日志对时间不敏感,即在产生事件日志的执行过程中从未发生版本演化^[4]。在过程挖掘中引入漂移的挑战是如何以时间依赖的方式表示行为变化。

早期的方法大多基于假设检验和统计测试,不仅需要巨大的特征空间,还需要人为选择合适的特征,同时还不易扩展。如 Bose 等^[5]将轨迹表征为特征向量进行统计测试, Maaradji 等^[6]直接对两个连续窗口之间的轨迹频率进行统计测试。这类方法保留了有损统计检验可靠性的低频行为,因此存在局限性。

基于相似性的漂移检测方法依赖于事件日志和提取的行为特征,如活动频率或活动之间的关系。Carmona 等^[7]从早期的数据中建立一个凸多面体来抽象表示包含多条轨迹的事件日志,并检查此多面体与最近轨迹的交点,但在检测到漂移后需要重新启动,扩展性不强。刘娜等^[8]使用流程一致性技术来检查从轨迹中提取的活动与发现的流程模型之间关系的相似性。该类方法在考虑不属于实际业务流程的行为时,可能会检测到虚假的漂移。

1.2 流程变体差异分析

从过程挖掘的角度来看,比较流程变体的方法可以用在概念漂移中来描述业务流程行为方面的差异。Adriansya 等^[9]基于对齐的方法将每个流程模型与相应事件日志进行比较来显示模型和流程执行之间的差异。Beest 等^[10]从对应的事件日志中计算行为的相关频率来识别两个事件日志之间的差异,但此方法只考虑了相对频率,存在局限性。Perner 等^[11]利用决策树来比较变体差异,由于只对其分支规则进行结构比较,容易造成高度过拟合的现象。Bolt 等^[12]对从子日志中提取的流程指标执行统计测试,但不能识别某些变化,也无法对有差异的流程行为进行可视化表示。Nguyen 等^[13]通过抽取日志中实体间的关系来构造关系图,并根据权重、公共节点和边进行比较,生成一个可视化的矩阵来突出显示两个过程变体之间的变化。

1.3 现存挑战

现有工作只强调漂移检测和变体差异分析这两个孤立任务而缺乏全局观念,漂移检测仅能单纯地定位发生变化的关键点,并不能对漂移点前后的流程行为变化进行详实描述,无法有效解决漂移定位与变体分析孤立存在的问题。因此,本研究提出一种基于漂移检测和流程变体差异分析的综合方法,在不需要额外的数据结构或过程模型的情况下,能够有效检测变化并解释变化的具体内容,帮助业务分析人员更好地理解流程行为并做出决策。

2 基于漂移检测的流程变体差异分析方法

2.1 基于漂移检测的流程变体差异分析方法

本研究提出的基于漂移检测的流程变体差异分析方法,整体框架如图 1 所示。首先在郑灿彬等^[14]所提方法的基础上进行改进,以事件日志为输入进行漂移检测;然后根据漂移检测的结果对事件日志进行分割,采用可视化方法将分割后的两个连续子日志的流程呈现出来;最后比较两个流程图中活动的发生顺序,标记并输出具有显著差异的活动结点和边。

2.2 漂移检测

过程挖掘领域的核心数据是事件日志,需要从事件日志中提取流程的相关信息,完成对漂移的检测和对流程差异行为的描述。本研究对文献^[14]进行改进,提出一种通过抽取日志中活动间行为上下文关系来检

测漂移的方法,整体架构如图 2 所示。

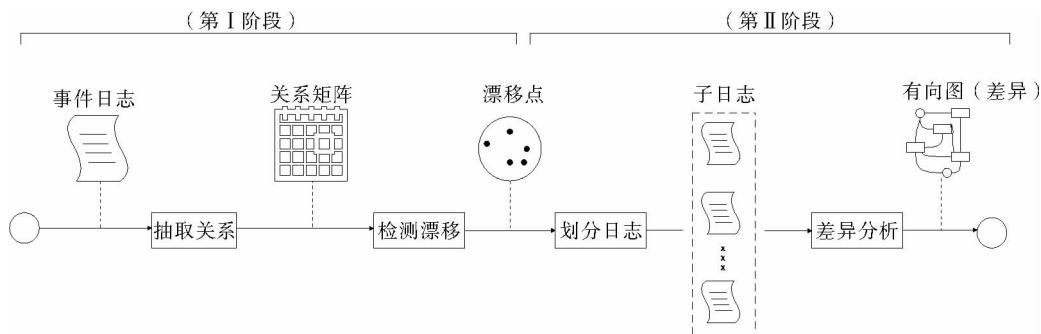


图 1 基于漂移检测的流程变体差异分析方法框架

Fig. 1 Framework of process variant analysis method based on drift detection

定义 1(行为上下文关系)

行为上下文关系^[15]是一对活动序列,事件日志 L 中的一组行为上下文关系表示为: $\beta_L \in \rho(A^* \times A^*)$, $\beta_L = \{(\sigma_l, \sigma_r) \in A^* \times A^* \mid \exists \sigma \in L, \sigma' \in A^* (\sigma_l \cdot \sigma' \cdot \sigma_r \in \sigma)\}$, 其中, A 为一组活动, A^* 为 A 上所有可能活动序列的集合, σ_l 和 σ_r 为 2 个活动序列, σ' 为则轨迹 σ 的子序列。

例如,在轨迹 $\sigma = \langle a, b, c, e \rangle$ 中, $\langle a, b \rangle$ 和 $\langle e \rangle$ 是围绕活动 $\langle c \rangle$ 的两个子序列,因

此,活动序列对($\langle a, b \rangle, \langle e \rangle$)是活动 $\langle c \rangle$ 的行为上下文关系。此处可设置关系长度,若关系长度设置为1,则活动 $\langle c \rangle$ 的行为上下文关系为($\langle b \rangle, \langle e \rangle$)。在确定了活动间的关系之后,可以通过抽取活动间的行为上下文关系来构造关系矩阵,进而表示事件日志中所有关系的变化情况。

定义 2(关系矩阵) 设 L 是一个包含 n 条轨迹的事件日志, 即 $L = \{\sigma_1, \dots, \sigma_n\}$ 。 \mathbf{D} 为一个关系矩阵, D_{ij} 表示矩阵 \mathbf{D} 中第 i 行第 j 列的元素, i 为日志中所包含的活动关系的有限集合, j 为轨迹的 ID。若关系 i 在轨迹 σ_j 上存在, 则 $D_{ij} = 1$, 否则 $D_{ij} = 0$ 。

定义 3(频率变化) 给定一种活动关系 i 和一个整数时间间隔 $[x, y)$, i 在 $[x, y)$ 上的频率变化情况为:

- 1) 总是存在, $D_{ij}=1$;
- 2) 从不存在, $D_{ij}=0$;
- 3) 有时存在, $D_{ij}=\{0 \cup 1\}$ 。

以事件日志 $L = \{abcd, acbd, abdc, adbc, acdb\}$ 为例, 表 1 为关系长度为 1 的行为上下文关系矩阵。根据关系矩阵的定义, 矩阵的每一行代表日志中一种活动的变化情况, 每一列代表每条轨迹中存在的所有活动关系。若活动关系 i 在时间间隔 $[x, y]$ 上总是存在或者从不存在, 表明关系 i 在此间隔内具有不变性, 将间隔长度预先设置为某个阈值。若在相邻两个时间间隔上的同一活动关系有变化, 即在给定的时间间隔内从一个值变为另一个值, 则说明流程可能在两个时间间隔的交点处发生了漂移, 此处的交点即为可能的漂移点。

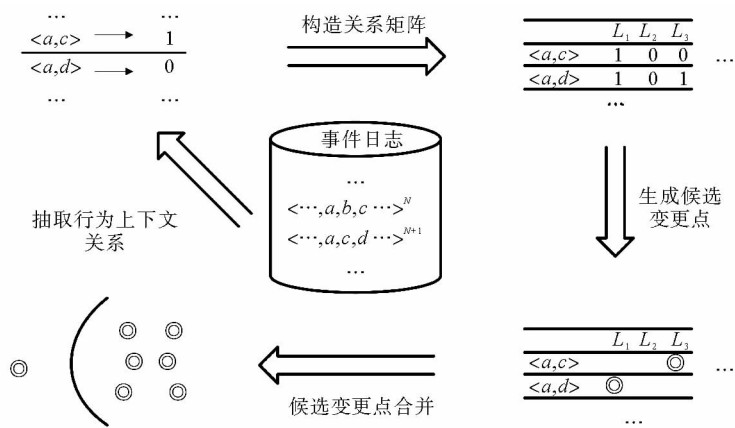


图 2 漂移检测方法

Fig. 2 Drift detection method

具体检测过程如图3所示。设置间隔长度为4,第一行为矩阵中的列号(轨迹ID),第二行为活动关系的变化情况。区间 $[1, 15]$ 以5为单位划分为3个区间,区间内活动关系经历了有时存在、总是存在、从不存在3个阶段。区间 $[1, 2)$ 和区间 $[3, 5)$ 距离长度小于4,不认为两个区间关系的过渡会产生漂移;区间 $[1, 5)$ 和区间 $[6, 10)$ 距离长度大于4,关系从有时存在变为总是存在,认为两个区间关系的过渡产生漂移;区间 $[6, 10)$ 和区间 $[11, 15)$ 距离长度大于4,关系从总是存在变为从不存在,认为两个区间关系的过渡产生漂移,故可能的候选漂移点为6和11。

表1 $L=\{abcd, acbd, abdc, adbc, acdb\}$ 的关系矩阵Tab. 1 The relation matrix of $L=\{abcd, acbd, abdc, adbc, acdb\}$

	σ_1	σ_2	σ_3	σ_4	σ_5
$\langle a, c \rangle$	1	0	0	0	0
$\langle a, b \rangle$	0	1	0	1	0
$\langle a, d \rangle$	0	0	1	0	1
$\langle b, c \rangle$	0	0	1	0	0
$\langle b, d \rangle$	1	0	0	0	0
$\langle c, b \rangle$	0	0	0	0	1
$\langle c, d \rangle$	0	1	0	0	0
$\langle d, c \rangle$	0	0	0	1	0

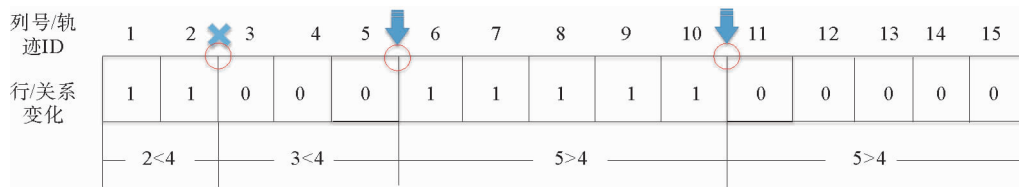


图3 候选漂移点检测

Fig. 3 Candidate drift point detection

在每个给定的活动关系上进行检测得到的候选漂移点不一定是真正意义上的漂移点,需要对候选漂移点进行聚类,以获得大概率事件下整个流程的漂移点。该聚类方法依赖于密度聚类算法,通过密度聚类将检测到的漂移点聚成若干点簇,按照簇中点的数量由大到小排序,簇中点的数量越多,该点簇越有可能代表最终的流程漂移点。通过对簇内所有漂移点进行聚合平均,获得表示流程漂移点的单一值,可用质心来表示。

示例过程如图4所示,基于密度聚类算法对候选漂移点进行合并,合并后的点簇 C_3 比点簇 C_2 包含更多的候选漂移点,说明点簇 C_3 更有可能代表一个真正的流程漂移点,此处点簇 C_3 用质心来表示。下面给出漂移检测的算法。

设事件日志中包含的轨迹有 n 条,活动关系数有 m 个,轨迹的平均长度为 k 。算法1中抽取行为上下文关系并转换成关系矩阵的复杂度为 $O(nk)$ 。由于候选漂移点的数量不会超过 mn/s ,因此,检测候选漂移点并对候选漂移点进行合并生成模型漂移点的复杂度为 $O(mn) + O(mn/s) = O(mn)$ 。综合来看,整个漂移检测算法的时间复杂度为 $O(nk) + O(mn)$,稍低于文献[14]中的时间复杂度 $O(nk^2) + O(mn)$ 。

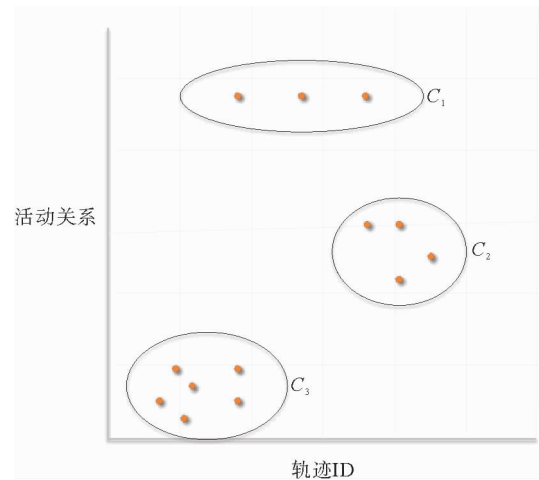


图4 候选漂移点合并

Fig. 4 Candidate drift point merging

算法 1 漂移检测输入:事件日志 L , 关系长度 d , 最小时间间隔 s , 聚类半径 r ;输出:模型漂移点集合 $Drift$.

```

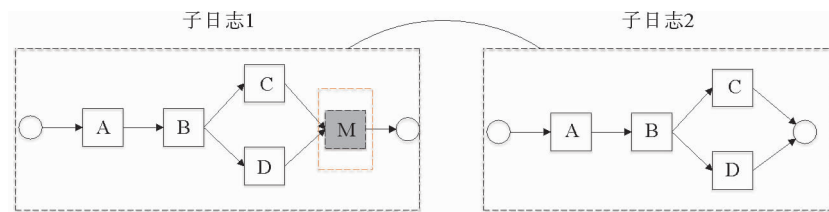
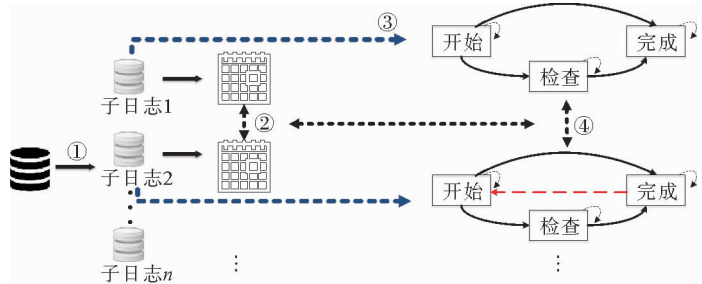
1)  $Drift \leftarrow \emptyset, List \leftarrow \emptyset, C \leftarrow \emptyset, res \leftarrow \emptyset, D \leftarrow \emptyset$ 
2) For  $i \leftarrow 1$  to  $\text{len}(\text{trace}) - 1$  do
3)   For key to  $res$ ,  $\text{key}()$  then
4)     if  $[\text{key}[i] \in \text{key}[i-1], \text{key}[i+1]]$  then
5)        $D \leftarrow \{D \cup res[i]\}$ 
6)   return  $D$ 
7) For  $j \leftarrow 0$  to  $D[i][n]$  do ( $i \leq n$ )
8)   if  $[D[i][j], D[i][j+1]] \in d$  and  $[D[i][j], D[i][j+1]] > s$  then
9)      $List \leftarrow \{List \cup D[i][j]\}$ 
10) return  $List$ 
11) For  $k \leftarrow 0$  to  $\text{clusters}[list, r]$  do
12)    $C \leftarrow \text{clusters}[k]$  then
13)    $\text{sort}(C)_{\text{max} \rightarrow \text{min}}$  then
14)    $Drift \leftarrow \{Drift \cup C\}$  then
15) return  $Drift$ 

```

2.3 流程变体差异分析

基于上述工作,在漂移检测的基础上,给出定位相邻变体间流程行为变化的算法,算法的整体架构如图 5 所示。①将原始事件日志按照检测到的流程漂移点切割成多个连续的子日志;②通过比较子日志间关系矩阵的相似性来预判相邻变体间的差异;③通过绘制有向图的方式来描述每个流程漂移点前后子日志中活动间的发生顺序;④根据有向图中对应活动和活动顺序的一致性来判定变体间具有显著差异的行为并对其进行可视化。

将分割后的两个连续子日志依据活动发生的顺序生成有向图,再对有向图中的活动顺序进行比较,若无法匹配,则在有向图中对活动结点和活动顺序进行标记。其中,活动的发生顺序是指活动的跟随关系,在有向图中用箭头来表示,箭头的始端表示先发生的活动,箭头的尾端表示随后发生的活动。图 6 为两个变体间活动片段出现差异的场景示例。

**图 6 差异分析示例****Fig. 6 Examples of variance analysis****图 5 变体差异分析算法****Fig. 5 Variant difference analysis algorithm**

算法2展示了流程变体差异分析的核心部分。设事件日志中包含的轨迹有 n 条,活动关系系数有 m 个,则算法2的时间复杂度为 $O(mn)$ 。本研究方法的整体复杂度为 $O(nk)+O(mn)$ 。

算法2 漂移检测
输入:模型漂移点集合 $Drift$;
输出:变体间的矩阵相似度 δ , 变体间具有显著差异的行为 $Diff$.
1) $\delta \leftarrow 0, Diff \leftarrow \emptyset, D \leftarrow \emptyset, count \leftarrow 0$,
2) For $i \leftarrow 0$ to $\text{len}(Drift) - 1$ do
3) basic on $Drift[i]$ split L to $L_0, L_1, L_2, \dots, L_{n-1}$ then
4) $D_{0 \sim n-1} \leftarrow \{L_{0 \sim n-1}\}$ compare $[\delta_{0 \sim n-1}, \delta_{1 \sim n}]$
5) $count++$
6) For $j \leftarrow 0$ to $count$ do
7) found activity and $D[j]$ then
8) if activity! = $D[j]$ then
9) $Diff \leftarrow \{Diff \cup D[j]\}$
10) return $Diff$

3 实验分析

3.1 数据集及实验设置

为了模拟概念漂移的存在,使用ProM工具对Petri网模型生成仿真日志。实验基于PC Intel Core i5-4210 M 2-60 GHz CPU, 12 GB RAM 环境,使用Python语言实现。

利用文献[6]中已公开发表的数据集进行测试,图7为用于测试漂移的一个贷款申请业务流程^[6]模型,共包括15个活动、1个开始事件和3个结束事件,并展示了循环、并发和选择等控制流结构。数据集中共包含72个与贷款申请业务流程不同变体相关的合成日志。

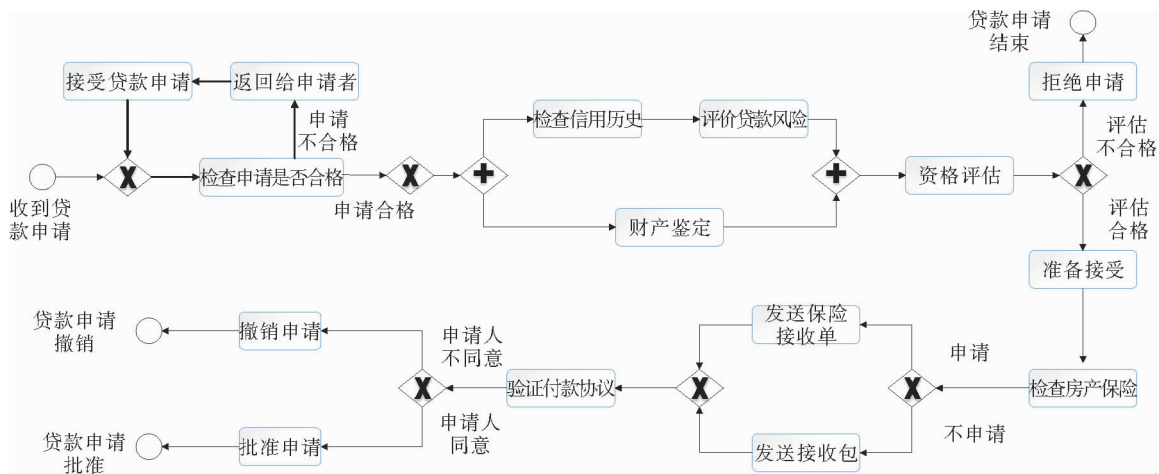


图7 贷款申请业务流程的参考模型

Fig. 7 Reference model of the loan application business process

表2为12种简单的控制流变更模式,为了进一步模拟更复杂的情况,将简单的变更模式分为插入(I)、重排序(R)和选择(O)3个类别,通过从每个类别中随机应用一个模式相互嵌套产生6个可能的综合变化模式:IOR、IRO、OIR、ORI、RIO、ROI。例如,综合模式IOR是通过插入(I)一个新活动,然后选择(O)该活动或现有活动,最后将整个模块放入重排序(R)结构中来获得。

在日志仿真过程中,先通过应用以上变更模式对基础模型进行修改得到新的模型,然后为每个修改后的新模型和基础模型仿真生成 5 组同等规模的事件日志,每组日志规模分别为 250、500、750、1 000,最后将每组日志按顺序交叉拼接得到混合事件日志。图 8 以 250 条轨迹的日志规模为例进行仿真,该事件日志混合了两个模型的日志,每个模型分别生成 5 组 250 条轨迹的日志,得到 10 组 250 条轨迹规模的日志,即最后的混合事件日志为 $L_{10,250}$ 。

3.2 评价指标

以日志拼接的方式生成混合日志,说明日志中发生漂移的位置是事先已知的。如果检测到的漂移点位于实际漂移点的某个较小邻域半径 R 内,则认为该检测结果正确, R 越小,检测的精度越高。为了评估检测的准确性,使用数据挖掘^[16]中针对漂移检测建立的既定指标,即以召回率和精确度的调和平均值衡量的 $F\text{-Score}$ 来度量方法性能:

$$precision = \frac{TP}{TP + FP}, \quad (1)$$

$$recall = \frac{TP}{TP + FN}, \quad (2)$$

$$F\text{-Score} = 2 \cdot \frac{precision \cdot recall}{(precision + recall)}. \quad (3)$$

其中: TP 表示正确检测到的漂移点数量, FP 表示误检的漂移点数量, FN 表示漏检的漂移点数量。

为了对检测到的具有显著差异的行为进行准确评估,设计一个基于矩阵相似性的方法来预判相邻变体的差异:

$$count = \sum_{i=1}^k \begin{cases} 0, & x_{i,j} = x_{i,j+1}; \\ 1, & x_{i,j} \neq x_{i,j+1}. \end{cases} \quad (4)$$

$$\alpha = 1 - \frac{count}{\sum_{i=1}^k x_i}. \quad (5)$$

其中: α 为矩阵相似度, j 为第 j 个矩阵, x_i 为矩阵中的某一活动关系, $x_{i,j}$ 为活动关系 x_i 在矩阵 j 中的成立情况。若活动关系 x_i 在相邻两个矩阵中成立情况不一致,则标记为 1,反之为 0。然后,统计成立情况不一致的活动关系在全部活动关系中所占比例,进而得到两个矩阵的相似度。

3.3 实验结果

为测试算法的效果,在不同规模的混合日志上($L_{10,250}, L_{10,500}, L_{10,750}, L_{10,1000}$)进行测试。图 9 为本研究算法在不同邻域半径 R 下对 12 种基本变化模式进行漂移检测的平均效果。由图 9 可以看出,本研究算法(简称本算法)的检测效果随着邻域半径 R 的增大而明显提升。聚类半径对检测效果的影响如图 10 所示,当聚类半径增大到超过每组日志的规模时,本该分开的点簇可能会聚合在一起,导致漂移点被漏检,从而影响召回率和准确率,故本算法在聚类半径较小时会取得好的效果。

表 2 12 种简单的控制流变更模式

Tab. 2 12 Simple control-flow change patterns

变更模式	具体描述	类别
Re	增加或删除片段	I
Cf	调整两个片段顺序	R
Lp	调整循环片段	O
Pl	将两个片段进行并发	R
Cb	跳过某个片段	O
Cm	将片段从选择分支中移除	R
Cd	将两个片段进行并发	R
Cp	重复某个片段	I
Pm	将某个片段从并发分支中移除	I
Rp	替换某个片段	I
Sw	交换两个片段位置	I
Fr	改变分支频率	O

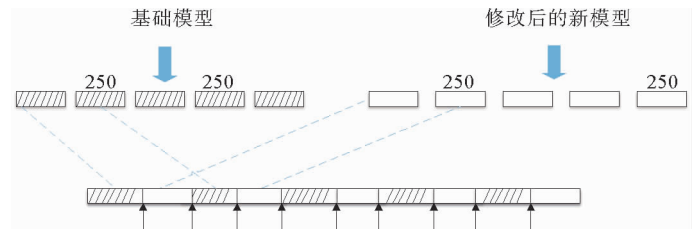


图 8 日志仿真过程

Fig. 8 Log simulation process

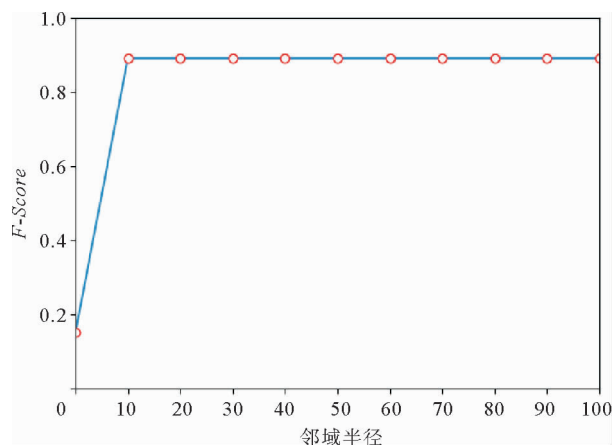


图9 不同邻域半径对结果的影响

Fig. 9 The effect of domain radius on the results

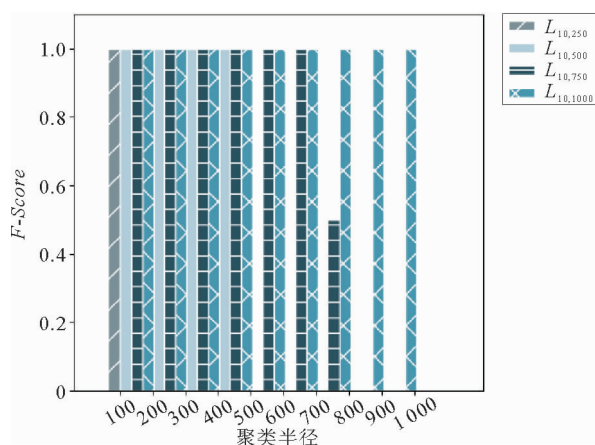


图10 不同聚类半径对结果的影响

Fig. 10 The influence of clustering radius on the results

此外,本算法是在文献[14]的基础上改进得到的,在保证漂移检测准确性的前提下,对本算法和文献[14]在时间效率上进行对比,见图11。可见,本算法的时间效率优于文献[14],进一步验证了二者在时间复杂度上的差异。

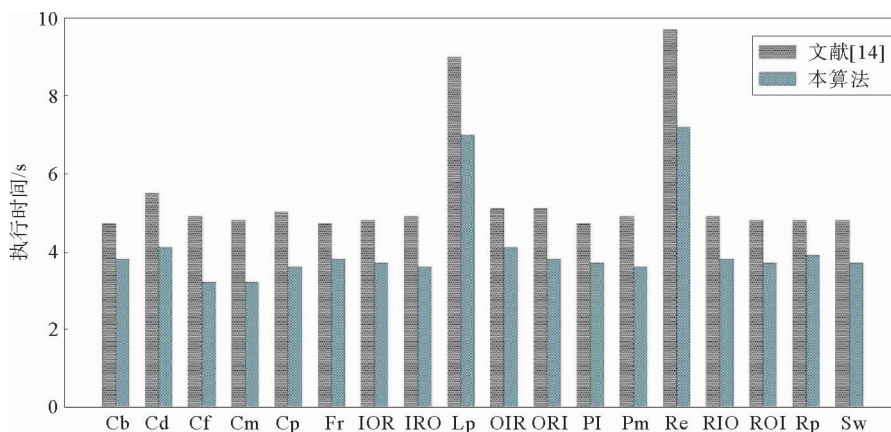


图11 时间效率对比

Fig. 11 Time efficiency comparison

漂移前后相邻变体间行为的差异能够体现流程的演变,故利用漂移前后相邻变体的相似度进行预判,能大概率的度量流程变体之间行为程度的差异。将聚类半径 R 固定为 10,在 ORI 模式下得到的混合日志 $L_{10,1000}$ 中检测出来的流程变体矩阵之间的相似程度见表3。

表3 $L_{10,1000}$ 中相邻流程变体矩阵的相似度Tab. 3 Similarity of the adjacent process variant matrix in $L_{10,1000}$

相邻变体	$L_1 \& L_2$	$L_2 \& L_3$	$L_3 \& L_4$	$L_4 \& L_5$	$L_5 \& L_6$	$L_6 \& L_7$	$L_7 \& L_8$	$L_8 \& L_9$
相似度	0.72	0.72	0.60	0.41	0.53	0.72	0.81	0.63

对漂移前后流程变体间的差异行为进行检测和标注,能够更准确地把握流程行为的演变。图12为综合变化模式 ORI 下,混合日志 $L_{10,1000}$ 中检测出来的变体1和变体2的有向图,图13为两者具有显著差异行为的可视化示意图。

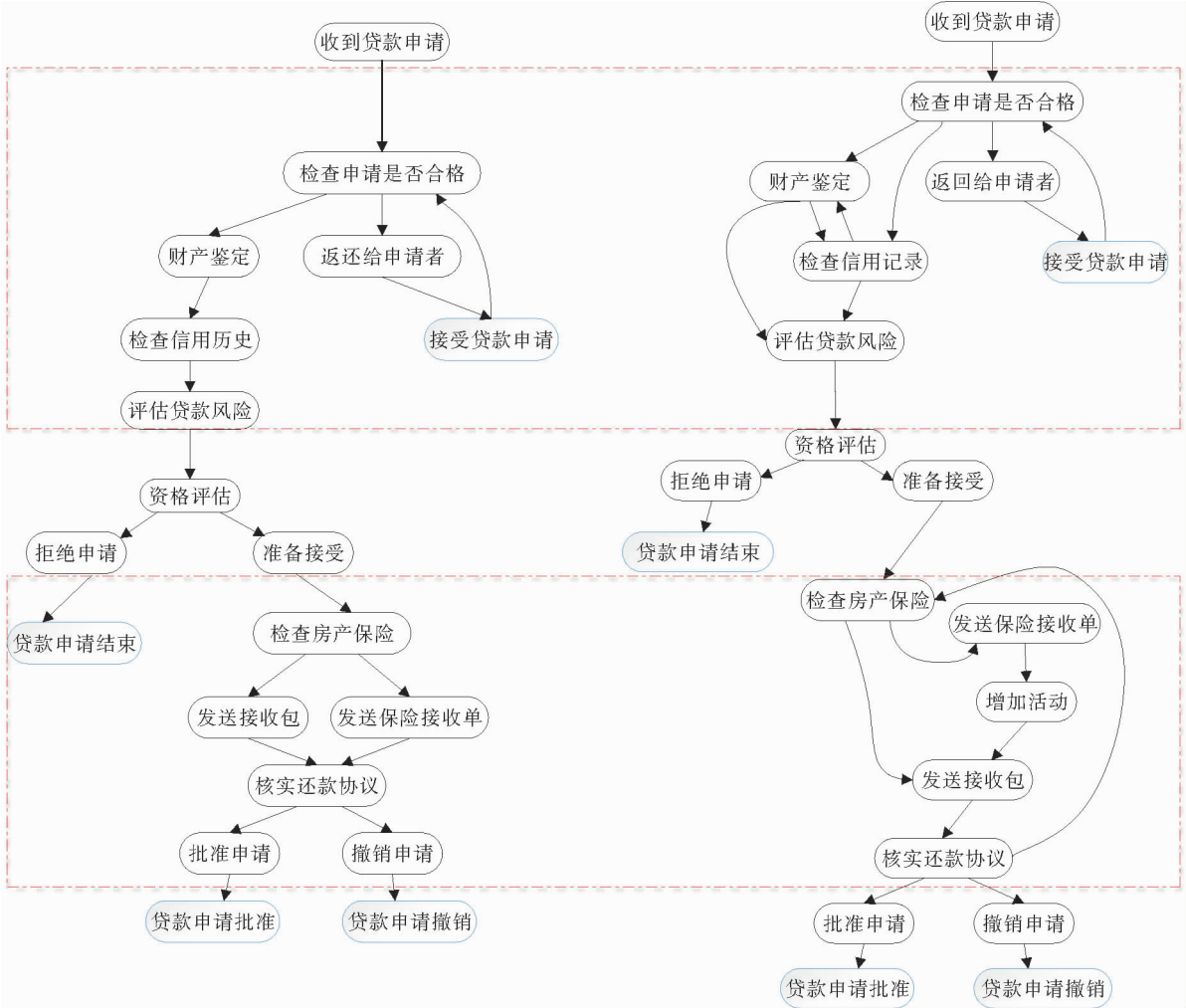


图 12 变体 1 与变体 2 的有向图
Fig. 12 Directed diagram of variant 1 and variant 2

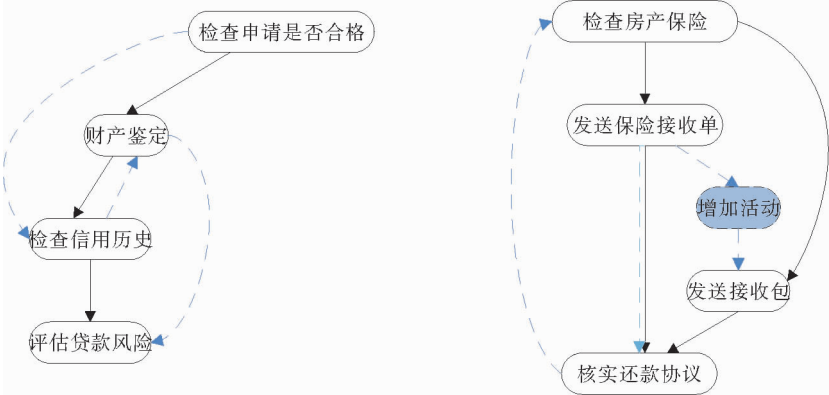


图 13 变体 1 与变体 2 差异行为的可视化
Fig. 13 Visualization of variant 1 and variant 2 differential behavior

根据上述实验可知,从浅层的检测漂移点到更深层次的定位及可视化描述能够揭露业务动态演化过程,可以帮助业务分析人员更好地进行流程配置和维护。

4 结论

通过抽取事件日志中活动间行为上下文关系的方法来检测漂移,不仅可以降低算法的时间复杂度,还能通过结果对漂移前后的变体流程进行差异检测。此外,根据所提评估方法对变体间行为差异的检测结果进行质量评估,间接证明了本算法的有效性。本算法主要针对离线场景下的概念漂移,未来可以从在线场景入手对漂移以及漂移后的流程演变进行分析或者预测。此外,针对更复杂的业务流程场景,如跨组织交互过程,尝试给出针对性的漂移检测算法。

参考文献:

- [1] AALST W M P. Process Mining: Discovery, Conformance and Enhancement of Business Processes[M]. Berlin: Springer, 2011.
- [2] BOSE R J, AALST W M P, ŽLIOBAITE I, et al. Handling concept drift in process mining[J]. Lecture Notes in Computer Science, 2011, 4(1): 391-405.
- [3] YESHCENKO A, CICCIO C D, MENDING J, et al. Visual drift detection for sequence data analysis of business processes [J/OL]. IEEE Transactions on Visualization and Computer Graphics, 2021, DOI: 10.1109/TVCG.2021.3050071.
- [4] 郑灿彬, 吴翔, 闻立杰, 等. 从事件日志中发现过程模型的渐变漂移[J]. 计算机集成制造系统, 2019, 25(4): 830-836.
ZHENG Canbin, WU Xuan, WEN Lijie, et al. Detecting concept drift of process models from event logs[J]. Computer Integrated Manufacturing System, 2019, 25(4): 830-836.
- [5] BOSE R J, AALST W M P, ŽLIOBAITE I I, et al. Dealing with concept drifts in process mining[J]. IEEE Transactions on Neural Networks and Learning Systems, 2014, 25(1): 154-171.
- [6] MAARADJI A, DUMAS M, ROSA M L, et al. Fast and accurate business process drift detection[C]// Proceedings of the International Conference on Business Process Management. Berlin: Springer, 2015: 406-422.
- [7] CARMONA J, GAVALDA R. Online techniques for dealing with concept drift in process mining[C]// Proceedings of the International Conference on Business Process Management. Berlin: Springer, 2012: 90-102.
- [8] 刘娜. 业务流程概念漂移检测与决策点挖掘算法的研究[D]. 北京: 北京邮电大学, 2019.
LIU Na. Research on process concept drift detection and algorithm for decision point mining[D]. Beijing: Beijing University of Post and Telecommunications, 2019.
- [9] ADRIANSYAH A. Aligning observed and modeled behavior[D]. Eindhoven: Eindhoven University of Technology, 2014.
- [10] BEEST N, DUMAS M, GARCIA-BANUELOS L, et al. Log delta analysis: interpretable differencing of business process event logs[C]// Proceedings of the International Conference on Business Process Management. Berlin: Springer, 2015: 386-405.
- [11] PERNER P. How to compare and interpret two learnt decision trees from the same domain[C]// Proceedings of the International Conference on Advanced Information Networking and Applications Workshops. Barcelona: IEEE, 2013: 318-322.
- [12] BOLT A, DE LEONI M, AALST W M P. Process variant comparison: Using event logs to detect differences in behavior and business rules[J]. Information System, 2018, 74: 53-66.
- [13] NGUYEN H, DUMAS M, ROSA M L, et al. Multi-perspective comparison of business process variants based on event logs [C]// Proceedings of the International Conference on Conceptual Modeling. Berlin: Springer, 2018: 449-459.
- [14] 郑灿彬, 闻立杰, 王建民. 基于可扩展活动关系的过程概念漂移检测[J]. 计算机集成制造系统, 2018, 24(7): 1589-1597.
ZHENG Canbin, WEN Lijie, WANG Jianmin. Process concept drift detection based on extensible activity relationship[J]. Computer Integrated Manufacturing System, 2018, 24(7): 1589-1597.
- [15] SANI M, ZELST S V, AALST W M P. Improving the performance of process discovery algorithms by instance selection [J]. Computer Science and Information Systems, 2020, 17(3): 927-958.
- [16] HOSS. A martingale framework for concept change detection in time-varying data streams[C]// Proceedings of the International Conference on Machine Learning. Bonn: Association for Computing Machinery, 2005: 321-327.

(责任编辑: 傅 游)