

一种基于过程挖掘的业务过程决策规则发现算法*

瞿 华

(北京科技大学 东凌管理学院, 北京 100083)

摘 要: 针对现有的决策点规则挖掘研究在挖掘时都只考虑了业务对象等过程外部对象的属性,而忽视了业务过程的内部属性——各决策点间的结构关系——对决策点的分支选择决策的影响,在深入研究过程内部属性提取方法的基础上,提出了一种基于过程挖掘的决策规则发现算法。该算法在挖掘决策规则时综合考虑过程外部对象属性和业务过程内部属性,从而可以更加全面、准确地挖掘决策点决策规则。实验结果证明,该算法能够有效挖掘业务过程决策规则,从而帮助用户更好地分析和理解实际业务过程。

关键词: 过程挖掘; 决策属性提取; 过程内部属性; 决策规则发现

中图分类号: TP391 **文献标志码:** A **文章编号:** 1001-3695(2012)06-2192-04

doi:10.3969/j.issn.1001-3695.2012.06.050

Business processes decision rules discovery algorithm based on process mining

Qu Hua

(Dongling School of Economics & Management, University of Science & Technology Beijing, Beijing 100083, China)

Abstract: Because the existing decision points' rule mining methods only focused on the relationship between business objects' attributes and decision points' branch decisions, they overlooked the impacts of the internal structural attributes of processes on the branch decisions. In order to solve this problem, this paper proposed a comprehensive decision rule discovery algorithm. By extracting the processes' intrinsic attributes, this algorithm considered both business objects' attributes and processes' internal structural attributes, so it could mine decision rules more accurately and completely. The algorithm was implemented and verified by experiments.

Key words: process mining; decision attributes extraction; processes' intrinsic attributes; decision rules discovery

现代企业出于管理的需要,在运营过程中会产生大量的业务活动日志,在这些日志中包含了丰富的关于企业业务过程的信息。如果能够将这些信息提取出来并以直观、易懂的形式提供给用户,就可以帮助企业管理者更好地认识、分析、评价和改进现有的业务过程,从而提升企业的执行能力和管理水平。过程挖掘试图通过对业务过程日志的挖掘,自动生成业务过程的执行流模型,从而帮助用户更好地理解业务过程的内在执行逻辑^[1,2]。经过多年研究,过程挖掘技术已经有了一批较为实用的算法^[3,4],并已经在实践应用中取得了一定的成功^[5-7]。在过程挖掘的基础上,找出过程中的决策点并针对其进行决策规则,可以发现过程的业务决策规则^[8-10]。但现有的基于过程日志挖掘的决策规则挖掘算法在挖掘决策规则时,都只考虑了过程外部属性,主要是业务对象的属性等对决策点的分支选择可能带来的影响,而忽视了过程内部属性,即过程的内部结构的影响。这一方面导致了分析的不全面,甚至可能会对用户产生误导;另一方面,在实际中能够得到的业务日志往往不包含过程外部属性信息,使分析根本无法进行。对此,本文提出了一种将所有从日志中能够得到的过程内外部属性信息全部参与挖掘的决策规则发现算法。

1 问题说明

过程挖掘通过对日志信息的分析来构造过程模型。如表1所示,为了保证挖掘算法能够最大限度地适用于各种形式的

日志,绝大多数挖掘算法仅要求日志中包含下列三项内容^[11]: a)事件所属的工作实例;b)执行事件的业务单元(任务标志); c)事件发生的顺序(处理时间)。

对表1中的日志进行整理,将其事件按照所属的工作实例分组,并按发生的先后顺序排列,可以得到表2形式的日志,与表1形式的日志相比更加直观。因此在对过程挖掘算法进行分析和说明时,多采用实例形式的日志。

表1 典型业务过程日志示例(部分)

工作实例标志	任务标志	处理时间
case 1	A	2010-5-8 08:15
case 2	A	2010-5-8 08:24
case 3	A	2010-5-8 09:30
case 2	B	2010-5-8 10:24
...	...	...

表2 实例形式的过程日志

工作实例	日志内容
case 1	AFG
case 2	ABCDEG
case 3	ABDCEG
case 4	ABCDEG
...	...

使用 α 算法^[12]对以上日志进行过程挖掘,就可以得到图1所示的过程模型。

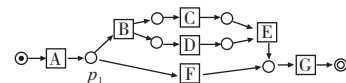


图1 挖掘得到的过程模型

通过分析过程挖掘得到的业务模型,可以发现其中存在着如图1中的 p_1 这样的决策点,其具有多个可选择的分支,实际执行时需要从中选择一条或多条来执行。从过程日志中找出决策点在作出决策时,业务过程所处的状态就可以利用决策挖掘算法得到决策点所采用的业务决策规则。

收稿日期: 2011-10-26; 修回日期: 2011-11-29 基金项目: 国家自然科学基金资助项目(71172169)

作者简介: 瞿华(1979-),男,江苏如东人,博士研究生,主要研究方向为管理信息化、数据挖掘、过程挖掘等(royqh1979@gmail.com)。

现有的基于过程日志的决策规则分析算法在考虑过程在决策点作出决策时的状态时,只考虑了过程的外部属性,即业务对象的属性等,忽视了过程的内部属性与过程状态及决策点决策的关系。而在实际中,内部属性在决策点的决策中往往也起着重要的作用。

例如,在图2(a)所示的过程中,决策点 p_2 的决策是由决策点 p_1 的决策决定的。如果 p_1 选择执行任务B,那么 p_2 只能执行任务E;如果 p_1 选择执行任务C,那么 p_2 只能执行任务F。再如,在图2(b)的过程中, p_2 选择执行E和F,是由 p_1 处循环的次数是否大于4来决定的。显然,这些决策规则只能从过程的内部属性,也就是过程的内部结构关系入手寻找,从外部属性中是无法挖掘出来的。

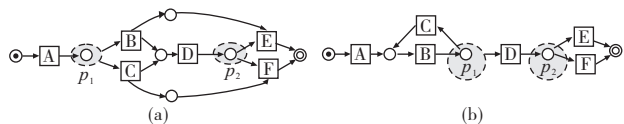


图2 内部属性与决策点决策

因此,本文提出了将过程模型与日志结合,从中提取过程内部属性信息,并将其与外部属性相结合,共同挖掘决策规则的算法。该算法与现有的算法相比,可以更加全面、准确地挖掘分支决策规则,从而帮助用户更好地理解过程的内在规律。

2 决策点及其决策属性

2.1 决策点及其分类

在过程挖掘基础上进行决策规则发现,第一步是要找出过程模型中的决策点。所谓决策点,就是过程模型中的执行流分支点。显然,决策点的具体定义与过程模型所采用的建模语言是密切相关的。目前,绝大多数过程挖掘算法都采用工作流网^[13]作为建模语言,它是Petri网的一个子集,具体定义如下:

定义1 工作流网。工作流网 N 为五元组 (P, T, F, i, o) 。其中, P 为全体库所集合, T 为全体变迁集合, F 为全体边集合, i 为输入库所, o 为输出库所。 $M_0 = \{i\}$ 为工作流网的初始配置。

图1和2中的过程模型都是工作流网模型。在工作流网模型中,决策点就是或一分支点,其定义如下:

定义2 决策点。对于工作流网 $N = (P, T, F)$,若有库所 $p \in P$ 满足 $|p \cdot| > 1$,则称 p 为 N 的一个决策点。 N 中的所有决策点的集合记为 $D(N)$ 。

从图形上看,决策点就是具有多个后继变迁的库所。例如图1中的 p_1 就是一个决策点,当该库所中存在标记(token)时,它的两个后继变迁B和F都处于被激发态,可以从中选择一个变迁触发执行。称这样的一个执行选择为决策点的一个分支。

定义3 决策点分支。若 p 为工作流网 N 的一个决策点,即 $p \in D(N)$,则对每个变迁 $t \in p \cdot$,称 t 为 p 的一个分支。 p 的所有分支集合记为 $b(p)$ 。显然, $b(p) = p \cdot$ 。

根据决策点分支是否构成循环,可以将决策点分支分成条件分支和循环分支两类。例如,图2(b)中的决策点 p_1 具有两个分支C和D,其中C是循环分支,D是条件分支。根据决策点的分支集合中包含的分支类型和数量不同,可以对决策点进行分类。

定义4 决策点类型。根据分支类型和数量的不同,可以

将决策点分为:

- 条件决策点,所有分支都是条件分支的决策点,如图1中的 p_1 ;
- 循环决策点,包含一个循环分支和一个条件分支的决策点,如图2(b)中的 p_1 ;
- 多重循环决策点,包含多个循环分支和一个条件分支的决策点,如图3(a)中的 p_1 ;
- 条件—循环决策点,包含一个或多个循环分支和多个条件分支的决策点,如图3(c)中的 p_1 。

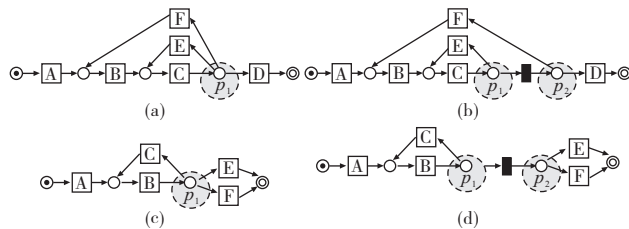


图3 复杂决策点及其转换

显然,循环决策点具有循环次数等结构性质,而条件决策点则没有。因此在寻找决策属性和进行决策挖掘时,有必要对这两类决策点区别。

至于更为复杂的多重循环决策点和条件—循环决策点,则可以采用引入隐含任务的方法在不改变触发序列等价性^[14]的前提下将其转换成条件决策点和循环决策点的组合。例如,图3(a)中的多重循环决策点 p_1 可以转换为两个循环决策点的组合,即图3(b)中的 p_1 和 p_2 ;图3(c)中的条件—循环决策点 p_1 可以转换为图3(d)中循环决策点 p_1 和条件决策点 p_2 的组合。

2.2 决策点的决策属性

决策点的决策属性就是过程在执行到决策点、要作出分支选择决策时,可能影响决策的各种属性。根据决策属性的所有者与过程的关系,可以将决策属性分为外部决策属性和内部决策属性两类。其中,所有属于过程外部对象的属性统称为外部决策属性,主要包括业务对象的各种属性、业务处理人员的各种属性、过程处理的时间和空间属性等;所有与过程内部执行流逻辑结构有关的属性称为内部决策属性,内部决策属性主要是分支选择属性和循环次数属性。

定义5 分支选择属性。它是指过程在执行中,经过决策点 p 时具体作出的分支选择构成的属性,记为 $c(p)$ 。 $c(p)$ 的取值集合为 $b(p)$ 。

根据定义5可知,过程执行每经过一个决策点,就具备了一个对应的分支选择属性。如图1所示,当过程执行经过 p_1 后,就具备了分支选择属性 $c(p_1)$,其取值由执行时实际选择的执行路径决定,或为变迁B,或为变迁F。

对于过程中循环决策点,还存在循环次数这一属性。

定义6 循环次数属性。该属性是指过程在执行中,经过循环决策点 p 的次数,记为 $n(p)$ 。 $n(p)$ 的取值范围是自然数集合。

分支选择属性和循环次数属性实际上是当前过程执行的历史状态,是过程内部逻辑结构的外在体现。使用它们参与决策属性挖掘,就可以找出过程内部决策点之间的相互制约关系。

2.3 决策点的决策属性提取中的问题

根据决策点及其决策属性的定义,可以直接从日志中提取决策属性。文献[8]已经详细讨论了外部决策属性和选择分

支提取的方法,以及隐含任务、重复任务等问题对选择分支提取的影响及解决办法,这里不再赘述。本文将着重讨论决策点的复合关系、决策点属性的更新以及循环分支的识别三个问题。

2.3.1 决策点复合关系及其处理

本文使用内部决策属性参与过程挖掘的目的是要找出过程内部决策点之间的相互制约关系。在这些关系中,有一类是决策点具有共同的后继变迁。例如,图4(a)中的 p_1 和 p_2 具有共同的后继变迁B;而图4(b)中的 p_1 和 p_2 具有相同的后继变迁集合 $\{A, B\}$ 。决策点间的这种关系称为决策点复合关系。

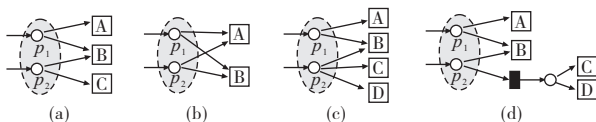


图4 决策点复合关系

显然,具有复合关系的两个决策点,其分支选择决策是由两个决策点共同作出的。例如,图4(a)中的 p_1 和 p_2 ,要么 p_1 执行A, p_2 执行C,要么 p_1 和 p_2 共同执行B;图4(b)中的 p_1 和 p_2 ,要么都选择执行A,要么都选择执行B。因此,在提取决策属性和进行决策规则挖掘时,应该将具有复合关系的多个决策点合并起来当做一个大的决策点来处理,其决策属性集合是所有组成决策点的决策属性的合集。

对于更加复杂的如图4(c)中的复合关系,则可以利用引入隐含任务的方法将其转换为由简单复合关系和简单决策点组成的结构,然后再进行处理。

2.3.2 决策属性的更新和记忆

在处理包含循环的过程模型时,会遇到决策属性更新的问题。例如,由于循环结构的存在,图5中的过程在执行时可能会多次经过决策点 p_1 ,其每次对应的分支选择属性 $c(p_1)$ 都可能会出现变化。

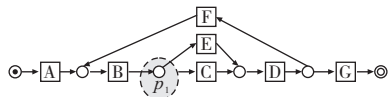


图5 循环与决策属性的更新

为了保证决策挖掘的准确性,如果假设过程遵循局部作用原则,那么每次属性变化时直接用新属性值更新属性值即可;否则,就需要在更新属性的同时也记忆旧属性值。因此,本文在算法中引入控制参数 r ,表示同一属性的最大记忆深度,取值范围为非负整数。如果 $r=0$,就表示不记忆旧属性值。

2.3.3 循环分支的识别

根据循环分支点和一般条件分支点的定义,它们的区别在于是否具有循环分支。因此,需要一种能够识别出循环分支的方法,以有效挖掘循环分支点的决策规则。由于复杂模型中存在的循环嵌套等问题,导致循环分支的准确识别成为了一个难题。目前尚未发现一种有效区别循环分支和普通分支的算法。因此,本文采用分析人员手工标志循环分支和循环起止点的方法来解决这一问题。

3 基于过程挖掘的决策规则挖掘算法

3.1 决策点分析算法

根据前面的讨论,决策点分析算法的目的是从给定的过程模型中找出决策点并进行必要的处理。

算法1 决策点分析算法。已知 workflow net $N = (P, T, F)$, 则决策点分析 DPAnalysis 算法步骤如下:

- 找出所有满足 $|p \cdot l| > 1$ 的库所 p , 加入决策点集 $D(N)$;
- 由用户手工标记出循环分支;
- 利用循环分支信息找出所有的循环分支点;
- 使用引入隐含变迁的方法化简复杂分支点;
- 找出所有具有共同后继变迁的决策点, 并使用引入隐含变迁的方法简化复杂的复合关系决策点。

3.2 决策点决策属性提取算法

算法2 决策点决策属性提取算法。已知 workflow net $N = (P, T, F)$ 和决策点集 $D(N)$, W 为 N 的日志集, 则决策点分支属性提取的 attributesGen 算法步骤如下:

- for each $\sigma \in W$
- $M \leftarrow \emptyset; A \leftarrow \emptyset;$
- for each $t \in \sigma$
- if $(\exists p \in D(N)) \wedge (p \in M)$
- 根据 determine decision classes 算法找出 $c(p)$ 的值, 生成一条决策样本: $p: A \Rightarrow c(p)$
- updateAttr($A, \{c(p)\}$)
- if p 是循环决策点
- $n(p) = n(p) + 1$
- updateAttr($A, \{n(p)\}$)
- end if
- end if
- $M \leftarrow M - \{t\} + \{t\}$
- updateAttr($A, A(t)$)
- end for
- end for

其中 determine decision classes 算法来源于文献[8], 其作用是找出过程执行实例在给定决策点处作出的决策选择; updateAttr 算法的作用是更新决策属性集合, 定义如下:

算法3 决策属性集合更新算法。已知 A 为决策属性集合, S 为要更新的决策属性集合, 则决策属性集合更新 updateAttr 算法的步骤如下:

- for each $a = (k, v)$ in S
- if $k \notin \text{key}(A)$
- $A \leftarrow A \cup \{(k, v)\}$
- else
- if $(r > 0)$
- if A 关于 k 的记忆数量少于 r
- 增加一条关于 k 的记忆
- else
- 滚动更新一条关于 k 的记忆
- end if
- end if
- end if
- end for

3.3 基于过程挖掘的决策规则挖掘算法

算法4 基于过程挖掘的业务过程决策规则挖掘算法。已知过程日志, 则基于过程挖掘的业务过程决策规则挖掘算法的步骤如下:

- 使用 α 算法等过程挖掘算法挖掘过程日志得到过程模型 N ;
- 使用 DPAnalysis 算法找出 N 中的决策点;
- 使用 attributesGen 算法生成决策规则样本;

d) 使用 C4.5^[15] 决策规则挖掘算法挖掘决策规则。

4 分析与验证

4.1 算法分析

已知过程日志 W 和过程模型 $N(P, T, F)$, 设 m 为日志 W 中所含日志实例数, n_1 为最长的日志实例所包含的任务数, $n_2 = |T|$ 为过程模型 N 中的变迁总数, $n_3 = |P|$ 是 N 中变迁数量, n_4 为最大外部属性数量。决策点分析的 DPAnalysis 算法的复杂度主要取决于决策点复合关系的寻找。构造共同后继关系矩阵的算法复杂度为 $O(n_3^2)$; 在共同后继关系矩阵基础上寻找具有复合关系的决策点集合的复杂度为 $O(n_3^2)$, 故 DPAnalysis 算法总体复杂度为 $O(n_3^2)$ 。

在决策点分支属性提取的 attributesGen 算法中, determine decision classes 算法的复杂度为 $O(n_3 \times n_2^2)$, updateAttr 算法的复杂度为 $O(n_2 \times (n_3 + n_4))$ 。所以 attributesGen 算法的复杂度为 $O(m \times n_1 \times n_2 \times (n_3 \times n_2 + n_4))$ 。

4.2 实验验证

本文在 ProM 平台^[16] 上以分析插件形式实现了相关的算法。其中, 过程挖掘算法直接使用 ProM 平台中的 α 算法模块; 决策规则挖掘算法采用 Weka (<http://www.cs.waikato.ac.nz/ml/weka>) 软件包中的 C4.5 算法模块。实验使用了包括图 2、3 和 5 中的五个模型在内的 10 个手工构造的过程实例进行。

图 6(a) 是实验中所使用的一个复杂过程实例。由于其结构中包含重复任务, 使用 α 算法直接挖掘其日志, 得到的过程模型如图 6(b) 所示。比较 (a) 和 (b) 两个模型可以清楚地看到, α 算法未能准确地挖掘出过程的实际逻辑结构。

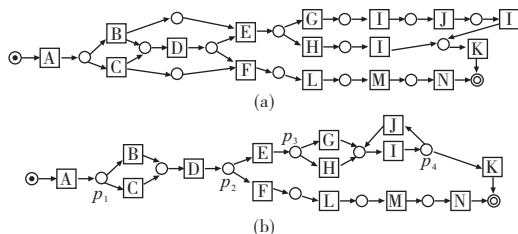


图6 实验验证使用的实例之一

结合日志对图 6(b) 中的模型使用本文提出的决策规则挖掘算法进行决策规则分析, 得到的包含内部决策属性的决策规则如表 3 所示, 这些规则很好地说明了算法未能发现的过程内在逻辑。对其他过程实例的实验验证也都得到了类似的结果, 从而证明了本文提出的算法的有效性。

表3 挖掘得到的包含内部决策属性的决策规则

决策点	决策规则
p_2	如果 p_1 执行 B, 则执行 E;
	如果 p_1 执行 C, 则执行 F;
	如果 p_3 执行 H, 则执行 K;
p_4	如果 p_3 执行 G 且循环次数小于 2, 则执行 J;
	如果 p_3 执行 G 且循环次数大于等于 2, 则执行 K;

5 结束语

本文在分析现有的业务过程决策规则挖掘算法缺陷的基础上, 提出了过程内部属性的概念及其提取算法, 并提出了在其基础上进行的基于过程挖掘的业务过程决策规则的算法。该算法适用于企业中的复杂业务过程分析问题, 它能够更加全

面、深入地挖掘业务过程的决策规则, 帮助用户更好地理解业务过程的内在运行机制。本文通过理论分析和实验, 验证了该算法的有效性和正确性。由于该算法采用了人工标志的方法来识别循环起止点, 在分析过程中仍然需要人工干预。因此, 下一步的研究方向是寻找一种自动化的循环起始点分析算法, 从而进一步提高算法的实用性。

参考文献:

- [1] Van DONGEN B F, De MEDEIROS A, WEN L. Process mining: overview and outlook of Petri net discovery algorithms [M] // JENWEN K, AALST W M. Trans on Petri Nets and Other Models of Concurrency II. Berlin: Springer-Verlag, 2009: 225-242.
- [2] Van der AALST W M P, Van DONGEN B F, HERBST J, et al. Workflow mining: a survey of issues and approaches [J]. Data & Knowledge Engineering, 2003, 47(2): 237-267.
- [3] WEN Li-jie, Van der AALST W M P, WANG Jian-min, et al. Mining process models with non-free-choice constructs [J]. Data Mining and Knowledge Discovery, 2007, 15(2): 145-180.
- [4] Van der AALST W M P, RUBIN V, VERBEEK H M W, et al. Process mining: a two-step approach to balance between underfitting and overfitting [J]. Software and Systems Modeling, 2010, 9(1): 87-111.
- [5] Van der AALST W M P, REIJERS H A, WEIJTERS A J M M, et al. Business process mining: an industrial application [J]. Information Systems, 2007, 32(5): 713-732.
- [6] BLUM T, PADOY N, FEUNER H, et al. Workflow mining for visualization and analysis of surgeries [J]. International Journal of Computer Assisted Radiology and Surgery, 2008, 3(5): 379-386.
- [7] LAU H C W, HO G T S, ZHAO Y, et al. Development of a process mining system for supporting knowledge discovery in a supply chain network [J]. International Journal of Production Economics, 2009, 122(1): 176-187.
- [8] ROZINAT A, Van der AALST W M P. Decision mining in business processes, Working Papers No. 164 [R]. Eindhoven, Netherlands: Eindhoven University of Technology, 2006.
- [9] ROZINAT A, Van der AALST W M P. Decision mining in ProM [C] // Proc of the 4th International Conference on Business Process Management. Berlin: Springer-Verlag, 2006: 420-425.
- [10] 高昂, 杨扬, 张静乐, 等. 基于工作流日志的决策规则挖掘研究 [J]. 计算机应用研究, 2009, 26(11): 4104-4107.
- [11] Van der AALST W M P, WEIJTERS A J M M. Process mining: a research agenda [J]. Computers in Industry, 2004, 53(3): 231-244.
- [12] Van der AALST W M P, WEIJTERS T, MARUSTER L. Workflow mining: discovering process models from event logs [J]. IEEE Trans on Knowledge and Data Engineering, 2004, 16(9): 1128-1142.
- [13] Van der AALST W. The application of Petri nets to workflow management [J]. Journal of Circuits, Systems and Computers, 1998, 8(1): 21-66.
- [14] ROZENBERG G, ENGELFRIET J. Elementary net systems [C] // Lectures on Petri Nets I: Basic Models, Advances in Petri Nets. London: Springer-verlag, 1998: 12-121.
- [15] QUINLAN JR. C4.5: programs for machine learning [M]. San Francisco: Morgan Kaufmann, 1993.
- [16] Van DONGEN B F, De MEDEIROS A K A, VERBEEK H M W, et al. The ProM framework: a new ERA in process mining tool support [C] // Proc of the 26th International Conference on Applications and Theory of Petri Nets. Berlin: Springer-Verlag, 2005: 444-454.