结合 DL-safe 规则发现日志本体频繁模式的方法

孙 明,陈 波,周明天
(电子科技大学计算机科学与工程学院,四川成都 610051)

摘 要: 为发现语义 Web 使用记录中所蕴含的有效信息,本文提出了一种挖掘日志本体频繁 Web 访问模式的方法.该方法引入应用访问规则集和观察集分别表示日志信息动态变化的语义规则和使用事实,并在 DL 安全的限定下将日志本体和应用访问规则集相结合构成一个推理过程可判定的混合知识库.在此基础上,利用日志本体中事件整分关系的语义构建访问模式学习的事务模型,并采用 ILP 的方法学习生成频繁用户访问模式树,解决了推理访问模式中非描述逻辑原子的问题.实验结果表明该方法的可用性和有效性.

关键词: 语义网使用挖掘; 日志本体; 频繁 Web 访问模式; DL-safe 规则; 归纳逻辑编程
中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2010) 02-0376-06

An Approach for Discovering Frequent Patterns from Log Ontologies with DL-safe Rules

SUN Ming, CHEN Bo, ZHOU Ming-Tian
(*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan* 610051)

Abstract: In order to discover the useful information from semantic Web usage records, we present an approach for mining the frequent Web access patterns from log ontologies. This method adopts application access-rules to represent the dynamic semantics rules of user-access and adopts observations to represent the usage facts. With the restriction of DL-safety, it combines log ontologies and application user-access rules into a decidable hybrid knowledge base. The transaction mode of access-pattern learning can be extracted form the semantics of the part-whole relations between events in log ontologies. A frequent Web access-pattern tree can be generated by an ILP method from the hybrid knowledge base. This method also solves the problem of reasoning the Web access – patterns with non-DL atoms. The experimental results show that this method is effective and it is quite feasible to solve practical problems.

Key words: semantic Web usage mining; frequent Web access pattern; log ontology; DL-safe rule; inductive logic programming (ILP)

1 引言

频繁模式发现是语义 Web 使用挖掘[1]中的关键过程,它从大量的使用数据中提炼出频繁的用户行为模式,挖掘的结果能够帮助管理者分析相关的设计和商业策略,从而获得商业价值的提升.传统挖掘算法采用 Web 日志文件作为挖掘集,而本体(Ontology)作为一种在语义和知识层面上的概念建模工具可以表示更丰富的背景知识,因此以本体为基础的频繁模式发现能更有效地发现用户访问的潜在语义.ONTOLOGGING[2]生成的日志本体(Log Ontology)就是一种描述 Web 使用信息的应用本体,它以事件为基础详尽地描述了用户使用站点的行为和策略语义.

本体以描述逻辑(Description Logic, DL)为基础,拥有相对丰富的概念定义,但缺乏强大的规则推理能力.因此本体之上的频繁模式挖掘系统,如 SPADA[3]和 SEMINTEC[4],常常是将本体与基于规则的逻辑相结合构成混合知识库,利用规则强大的逻辑推理能力弥补本体的不足.SPADA 以[5]为基础,生成的模式只能适用于在知识库中明确存在的实例,且不支持角色对应的二元谓词.SEMINTEC 基于 OWL – DL,它将生成的模式用 DL-safe 规则[6]表示,以 KAON2[7]为逻辑推理机有效地提高了本体之上模式发现的质量和效率,但它不支持对附加规则集和观察集的推理.

收稿日期:2008-12-24;修回日期:2009-02-16
基金项目:国家"十一五"科技支撑计划重大资助项目(NO.2006BAH02A0407);国家自然科学基金(NO.90604033)

Web 日志系统是一个动态变化的数据集,随着站点访问量的增加而不断有新的使用信息加入,这些信息刻画了日志本体作为背景知识所不具有的动

态访问语义,因此发现日志本体频繁 Web 访问模式要求不仅要能对概念和角色进行建模和推理,而且要能处理实时动态变化的应用语义.因此 SPADA 和 SEMINTEC 都难以直接应用于语义 Web 使用知识的处理.针对使用知识动态的特点,本文提出了一种基于日志本体频繁 Web 访问模式发现的方法,引入应用访问规则集和观察集分别表示日志信息动态变化的语义规则和使用事实,并在 DL 安全的约束下将日志本体和应用访问规则集结合为混合日志知识库,利用日志本体中事件整分关系的语义构建频繁模式发现事件事务模型,采用 ILP 的方法学习生成频繁 Web 访问模式树.该方法提升了访问模式的表达力,同时利用日志本体事件关系特定的语义减小模式集的构建规模,实现对模式中非描述逻辑原子的推理,提高发现效率和结果的有效性.

2　混合日志知识库

2.1　日志本体

日志本体是语义 Web 使用知识结构化表示[8],以事件为核心概念描述了用户访问站点的语义行为.根据访问目的和访问策略的不同,事件从抽象语义上分为原子事件(AtomEvent,EA)和复合事件(ComplexEvent,EX).原子事件表示用户一次具体的访问行为,而复合事件是由原子事件构成的有序序列,在更抽象的层次上可以被认为是事件层次体系结构中的一种语义访问策略.日志本体采用 W3C 推荐的 OWL-DL 语言描述,其描述逻辑基础从 SHOIN(D)归约为 SHOIN(D),为了方便算法的推理,本文统一使用描述逻辑的方式来表示日志本体.

定义 1　日志本体 给定使用领域 D,L 是 D 上的逻辑语言.D 上的日志本体定义为一个五元组,$LO:=\{E,\leqslant_E,R,F,A\}$.其中:$E$ 是事件集合;$\leqslant_E$ 是 E 上的继承关系;R 是 E 上非继承关系集合;F 是关系函数,$F:E\times E\rightarrow R$;A 是基于 L 的推导公理集.

2.2　结合日志本体与 DL-safe 规则的混合知识库

日志本体描述了站点固有的使用背景知识,但是不能表示用户访问过程中动态变化的应用语义.为此,本文引入应用访问规则集,采用 Datalog 子句来描述这种变化的访问语义,例如:在某电子商务站点使用语义中,*BuyProduct*(x)表示用户购物事件,使用非描述逻辑原子 *searchTimes*(x,n)表示某次访问策略中使用了 n 次查询服务,其中 *searchTimes* 是 *Datalog* 谓词,而查询次数少于三次的购买事件定义为直接购物事件 *DirectBuy*(x),可用以下规则表示:

$$DirectBuy(x)\leftarrow BuyProduct(x),\ searchTimes(x,n),smallerThan(n,3) \quad (1)$$

结合应用规则的日志本体能满足日志系统动态增长的特点,可以提高系统推理能力,但容易引起推理过程的不可判定.为了避免这个问题本文采用 DL 安全[8]对结合进行限定,要求规则前件中的变量必须至少出现在一个非描述逻辑项的谓词中,确保所有规则使用的个体都被显式引入到知识库.满足 DL 安全限制的规则称为 DL-safe 规则.虽然 DL 安全对于规则的安全性限制较强,但在实际推理过程中可以通过技术手段为规则添加特殊的非描述逻辑原子 $O(x)$并在规则库中为对应个体 a 添加基原子 $O(a)$的方式来满足这种限制.

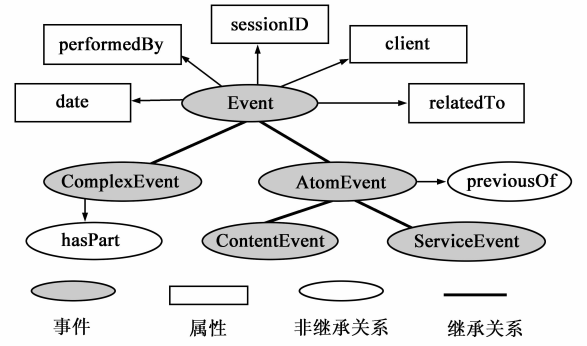


图1　日志本体结构片段

接下来定义基于 DL－safe^L 的日志本体混合知识库.DL－safe^L 是在描述逻辑语言 L(这里 L 限定为 SHIQ(D))的基础上满足 DL-safe 规则的一组选言 Datalog 语言,有符号集 N 和 C,其中 N 是谓词集,$N=N_C\cup N_R\cup N_P$,C 是常量集.特别地,N_C 中的谓词是概念,N_R 中的谓词是角色,N_C 和 N_R 中谓词统称为描述逻辑谓词(*DL* 谓词).N_P 为任意元非描述逻辑谓词符号集,称为 Datalog 谓词.由于日志本体中事件个体都是被命名的,因此常量集 C 中的个体都是实名个体(Named Individuals).

定义 2　DL－safe^L 日志本体知识库 $B=(KB,P)$,其中 KB 是对应于描述逻辑 L 的日志本体库,P 是由 DL-safe 规则 r 组成的应用访问规则集.规则 r 形如:

$$h_1(X_1)\leftarrow h_s(X_s)$$
$$a_1(Y_1),\cdots,a_m(Y_m)\&b_1(Z_1),\cdots,b_n(Z_n) \quad (2)$$

其中 $h_1,\cdots,h_s$ 是 Datalog 谓词或 *DL* 谓词,$a_1,\cdots,a_m$ 是 Datalog 谓词,$b_1,\cdots,b_n$ 是 *DL* 谓词,$X_1,\cdots,X_s,Y_1,\cdots,Y_m$ 是任意元项的序列,$Z_1,\cdots,Z_n$ 是一元或二元项的序列.

应用访问规则集 P 描述日志系统运行过程中动态变化语义规则.为了在 DL－safe^L 统一的规则下挖掘日志本体,需要将基于描述逻辑语言 L 的日志本体库 KB 等价地转化为选言 Datalog 规则库 $DD(KB)$,并在此基础上增加 DL-safe 应用访问规则集 P 构成一阶规则知

识库 $DD(KB) \cup P$. 文献[9]给出了 SHIQ(D)到 $DD(KB)$的转换过程.

3 基于 DL – safe^L 发现频繁 Web 访问模式

3.1 频繁 Web 访问模式发现的任务

复合事件在更高的层次上表示了用户的访问策略.发现日志本体频繁模式的任务是在特定访问策略中发现用户使用站点的频繁行为,即从复合事件 EX 出发找出与原子事件 EA 之间的频繁项,给定:

- 基于 DL – safe^L 的知识库 B;
- 基准事件 $E_{ref} \in EX$;
- 观察集 O, $B \cap O = \varnothing$;
- 最小支持度阀值 minsup.

从基准事件 E_{ref}出发找出所有满足条件的模式 H 计算其支持度 s,若 $s \geqslant$ minsup 称 H 为频繁模式,频繁模式发现的任务就是找出所有频繁模式构成的集合 FPS.

观察集 O 包含应用访问规则集 P 中对应的观察事实,观察 $o_i \in O$ 定义如下.

定义 3 给定 DL – safe^L 知识库 B,观察 $o_i \in O$ 是二元组$(q(e_i), A_i)$,其中 e_i 是事件个体,$q(e_i)$是 Datalog 基原子,A_i 是一组关于 e_i 的 Datalog 事实.

Web 访问模式 H 是发现基准事件 E_{ref}和任务相关事件之间联系的一组联合查询,其应答集由 E_{ref}个体组成.

定义 4 给定基准事件 E_{ref},关于 DL – safe^L 知识库 B 的 *Web* 访问模式定义 H 为具有以下形式的 DL-safe 规则:

$$q(X) \leftarrow a_1(Y_1), \cdots, a_m(Y_m) \\ \&E_{ref}(X), b_1(Z_1), \cdots, b_n(Z_n) \qquad (3)$$

其中 $a_1, \cdots, a_m$ 是 Datalog 谓词, $b_1, \cdots, b_n$ 是 DL 谓词, X 是特征变量.特别地, $H_{ref} = q(X) \leftarrow \&E_{ref}(X)$称为平凡模式. H 的应答是一个基置换 θ,若,则称 θ 是 H 关于知识库 B 的正确应答.应答集 answerset(H, E_{ref}, B)是 H 关于 B 的正确应答构成的集合.

定义 5 给定模式 $H \in$ DL – safe^L,其支持度为 H 与其平凡模式 H_{ref}应答集中基准事件个体数之间的百分比:

$$\text{support}(H, E_{ref}, B) = \frac{|answerset(H, E_{ref}, B)|}{|answerset(H_{ref}, E_{ref}, B)|} \qquad (4)$$

3.2 发现频繁 Web 访问模式的 ILP 方法

ILP 是一种有效利用背景知识的一阶逻辑学习方法,本文借助 ILP 的理论学习混合知识库 B 中的频繁模式,要求在保证一致性的前提下归纳生成模式 H,使得 H 与知识库 B 逻辑蕴含观察集 $O(B \cup HO)$并且 H 是频繁的.

复合事件与原子事件之间的整分关系是日志本体中一种重要的非分类关系,用 hasPart 来表示.这种整分关系下的复合事件描述了不同访问策略中访问行为的构成情况,其个体 ex 可以表示为 $ex = < ea_1, ea_2, \cdots, ea_n >$,这样就构成了频繁 Web 访问模式学习的事件事务模型,其中 ea_i 是原子事件个体,$1 \leqslant i \leqslant n$.这种模型下基准事件 E_{ref}限定为复合事件,本文从 E_{ref}出发采用构造频繁模式树 FPT 的方法归纳频繁模式.FPT 中从根节点出发到任意子节点的路径就是 E_{ref}的一个频繁访问模式.频繁模式集 FPS 则是由所有基准事件的 FPT 构成的森林.图 2 是一棵 FPT 的部分.

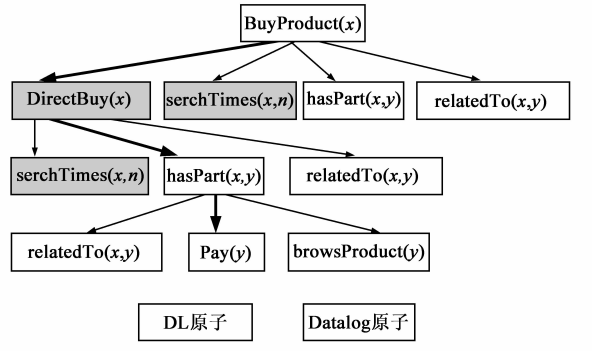


图2 频繁访问模式树

FPT 采用类似 FARMER 中定义的 trie 结构[10]. 节点 $N(x, y_n)$表示一个原子项,其中 N 是原子对应的谓词, x 是特征变量, y_n 表示普通变量集.构造 FPT 的规则如下:从根节点 $E_{ref}(x)$出发,采用广度优先的方式为每一个节点 $N(x, y_n)$扩展满足 ILP 要求的子节点,直到达到最大层次 MAXDEPTH,或没有新的频繁模式产生.为有效计算 $N(x, y_n)$的扩展节点,本文从日志本体中事件关系的语义出发定义容许谓词集 AP 缩减 FPT 的构建规模:$N(x, y_n)$的容许谓词集 AP_N 可以是 KB 中 DL 谓词以及 P 中 Datalog 谓词,包括依赖谓词(其对应原子与 $N(x, y_n)$至少有一个共享变量),右兄弟谓词(其对应原子是 $N(x, y_n)$的右兄弟节点)或 $N(x, y_n)$自身谓词,同时频繁模式是针对特定访问策略发现访问行为之间的频繁项,因此一棵 FPT 中除 $E_{ref}(x)$及衍生事件外不能包含其它复合事件.

从 AP_N 中选择谓词 A 构造其对应的原子 $A(x, y_a)$,若添加 $A(x, y_a)$构成的新模式 H 通过语义自由和概念分类冗余检测[11],且 H 与 B 相容于 O 且频繁,则将其添加为 $N(x, y_n)$子节点扩展 FPT.上述验证过程基于实例检查,对于知识库中 E_{ref}的每一个事件个体 e_i, o_i 是其对应的观察,若 H 关于 B 蕴含 o_i,则 e_i 是 H 的正确应答,加入应答集 answerset(H, E_{ref}, B),然后通过应答集计算支持度.下面给出 H 关于 B 蕴含观察 o_i 的定义.

定义 6 给定模式 $H \in DL - safe^L$,知识库 B 以及观察 $o_i \in O$, H 关于 B 蕴含 o_i 当且仅当 $B \cup H \cup A_i q(e_i)$.

引理 1　q 为 $DL-safe^?$ 知识库 $K=B\cup O$ 的基查询,若 Kq 当且仅当 Kq.

证明过程与文献[12]类似,这里不再赘述.

定理 1　给定模式 $H\in DL-safe^L$,知识库 B 以及观察 $o_i\in O$,若 H 关于 B 蕴含 o_i 当且仅当 $B\cup H\cup A_iq(e_i)$.

证明:H 关于 B 蕴含 o_i

$\Leftrightarrow B\cup H\cup A_iq(e_i)$　　(定义 5)

$\Leftrightarrow B\cup H\cup A_iq(e_i)$　　(引理 1)

因此验证 e_i 是 H 的正确应答就等价于是否能为查询$\leftarrow q(e_i)$找到关于 $B\cup H\cup A_i$ 一个正确的应答.扩展节点的算法如图 3 所示.

FPT 在扩展节点构造频繁模式的过程中引入了应用访问规则集中表示访问语义的 Datalog 原子,其对应的事实由相应的观察描述,因此在节点扩展过程中有两种方式可以完成 Datalog 原子的推理:将观察作为特殊的断言手工添加到混合知识库中,然后通过 KAON2 直接完成推理;或者在 DL 实例验证之前,先完成对模式中 Datalog 原子的消解.为了提高发现过程的自动化程度,本文选择后者,参考中的制约 SLD 否证[5],消除 Datalog 原子,然后将基化后的模式提交至 KAON2 完成 DL 原子查询.同时,因为事件个体都是有名个体,因此制约 SLD 消解过程不会出现匿名个体无法基化的问题.

expandNode($N(x,y_n)$,nodeLevel)

```
if nodeLevel < MAXDEPTH
    计算 N(x,y_n)容许谓词集 AP_N;
    foreach A∈AP_N do
      构造 A 对应的节点A(x,y_a);
      构建从根节点到 A(x,y_a)的模式 H;
      if H 通过语义自由和概念分类冗余检测 then
          foreach e_i∈E_ref do
            if H 中存在 Datalog 原子 then
              采用制约 SLD 否证消解 Datalog 原子;
      if e_i 和 H 满足 DL 推理机实例检测 then
              将 e_i 加入 answerset(H,E_ref,B);
        end for
      计算 H 的支持度 s;
      if s≥minsup then
          添加 A(x,y_a)作为 N(x,y_n)的子节点;
    end if
    end for
    foreach B(x,y_b)∈ N(x,y_n)的子节点 do
        expandNode(B(x,y_b),nodeLevel+1);
end for
end if
```

图 3　节点扩展算法

4　仿真实验

为验证文中提出方法的有效性和可靠性,本文以 Java 为编程语言实现方法原型并进行仿真实验.实验中本文的提出方法命名为 LOFPD(Log Ontology Frequent Pattern Discovery).测试环境采用 Intel Core2 2.2G,2G 内存,Windows 2003 server sp2,J2SDK1.5,推理机采用 KAON2(release 2008.6.29).测试数据集包括 PKDD CUP[8]和 OMS,均采用 OWL-DL 语言描述.其中 PKDD CUP 是 SEMINTEC 的主要测试本体,包括 60 个概念和 16 个角色,17941 个实例.OMS 是利用 ONTOLOGGING 和 Protégé 根据某手机电子商务网站 8 个月日志文件生成的日志本体,包括 58 个事件(包括 4 个复合事件)和 18 个角色,27723 个实例,应用规则 11 条,观察事实 923 个.测试集的选取原则是要求满足日志系统的特点:关系相对不复杂,但实例数据量较大.PKDD CUP 虽然不是专门的日志本体,但作为一种重要的本体测试集,其特点也基本满足本文测试要求.在上述测试集中对 LOFPD 和 SEMINTEC 进行对比测试,为保证测试一致性,SEMINTEC 采用 SAT 完整相容性测试.

对于测试集 PKDD CUP,设置 minsup = 0.2,MAXDEPTH = 7,基准概念为 CreditCard.为了测试引入应用访问规则集对算法效率和结果的影响,我们在 PKDD CUP 上附加规则集如下:

$$CardSlave(x)\leftarrow moreThan(y,5)\ \&client(x)\ hasCreditCard(y) \quad (5)$$

$$HighRiskClient(x)\leftarrow CardSlave(x)\ \&hasAgeValue(x,y)\ Above65(y) \quad (6)$$

同时为规则集添加对应的观察事实 80 个,测试结果如图 4 和表 1.

测试集 OMS 本身已经是包含应用规则库的混合知识库,设置 minsup = 0.2,MAXDEPTH = 6,基准事件为 BuyProduct.测试结果如图 5 和表 2.

性能测试表明:因为 SEMINTEC 不支持应用访问规则集,当测试数据只考虑本体集时,PKDD CUP 上 SEMINTEC 和 LOFPD 对同一个核心事件生成 FPT 的时间基本一致,因为两者对描述逻辑知识的推理均采用 KAON2;而在 OMS 上由于 LOFPD 采用了事件语义规则限制了容许谓词的规模,因此其生成 FPT 的时间要优于 SEMINTEC.当测试数据在本体集上附加规则集和观察集后,LOFPD 完成了对非 DL 原子的推理,虽然耗时稍长但是与无规则集上 SEMINTEC 和 LOFPD 的测试时间相比并没有增加计算复杂度,仍然是 EXPTIME.因为基于 DL-safe$^?$ 的日志本体混合知识库的描述逻辑和一阶规则分别为 SHIQ(D)以及选言 Datalog 程序,SHIQ(D)中实例验证的计算复杂度为 EXPTIME-完全[6],而

基于选言 Datalog 的 DL-safe 程序复杂度却是 EXPTIME[13],LOFPD 仅是在将模式提交到描述逻辑知识库进行实例验证前利用制约 SLD 否证的方法对满足 DL-safe 的选言 Datalog 规则推理消除 Datalog 原子,可以看成是两种逻辑上推理算法的直接联接,因此其计算复杂度仍限定在 EXPTIME 范畴中.当然在以后将描述逻辑基础扩展到 SON(D),解决标称事件以及数量限制等当前工作中的难题后,计算复杂度将上升到 NEXPTIME.

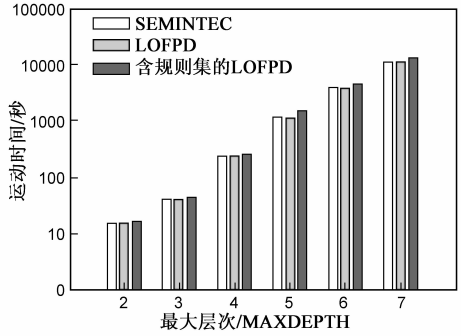


图4 PKDD CUP上SEMINTEC和LOFPD生成FPT的时间以及附加规则集后LOFPD生成FPT的时间

表 1 PKDD CUP 上 SEMINTEC 和 LOFPD 生成频繁模式的数量

FPT 层次	SEMINTEC	LOFPD	附加规则集 LOFPD
2	10	10	10
3	23	23	25
4	85	85	91
5	276	276	285
6	496	496	510
7	831	831	847

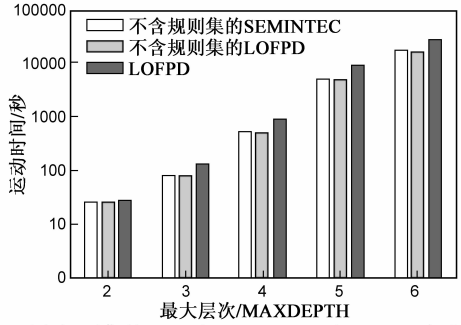


图5 不含规则集的OMS上SEMINTEC和LOFPD生成FPT的时间以及完整OMS上LOFPD生成FPT的时间

表 2 OMS 上 SEMINTEC 和 LOFPD 生成频繁模式的数量

FPT 层次	不含规则集 SEMINTEC	不含规则集 LOFPD	LOFPD
2	5	4	5
3	17	14	16
4	59	49	57
5	154	127	149
6	307	231	255

测试结果表明:在 PKDD CUP 上 SEMINTEC 和 LOFPD 生成 FPT 结果完全一致,因为两者均采用 trie 结构生成频繁模式,并且 SEMINTEC 不是专门的日志本体,FPT 生成过程不受 LOFPD 中基于事件语义规则构建节点容许谓词集策略的限制.但是在 OMS 上,LOFPD 利用日志本体事件语义规则缩小了容许谓词集的规模,摒弃了与访问策略无关的模式,因此生成的频繁模式数量少于 SEMINTEC,但更符合日志本体访问策略的表示.当两组测试集在附加规则集和观察集后,由于表示动态使用信息的 Datalog 原子被引入访问模式,因此 LOFPD 可以发现更多蕴含应用语义的频繁模式满足日志系统动态变化的特点.如 OMS 上无应用访问规则集时两种方法都可以发现模式:

$$q(x) \leftarrow \&BuyProduct(x), hasPart(x,y), serch(y), relatedTo(y,z), Nokia(z) \quad (7)$$

当加入应用访问规则集后只有 LOFPD 能发现具有动态应用语义的模式:

$$q(x) \leftarrow searchService(x,n), smallerThan(n,3) \&BuyProduct(x), hasPart(x,y), serch(y) \quad (8)$$

作为结论,LOFPD 引入了应用访问规则集增加了对日志系统动态使用语义的支持,在不增加本体库推理计算复杂度的基础上可以推理带非描述逻辑原子的模式,适合于语义 Web 使用挖掘中发现日志本体频繁模式的学习.

5 结论

为更有效地发现用户频繁访问模式,本文将日志本体和应用访问规则集相结合,在 DL 安全的限制下构成混合知识库作为挖掘的基础.应用访问规则集描述了站点用户动态变化的语义行为,弥补了日志本体动态访问知识表达不足的缺陷.通过日志本体事件整分关系语义构建了 Web 访问模式学习的事务模型,在此基础上借助 ILP 的理论和方法生成频繁访问模式树,在不提高计算复杂度的基础上提高了访问模式的表达力以及混合知识库推理的能力,挖掘的结果更加准确和有效.下一步我们将定义用户在不同商业领域的站点访问策略,利用该学习规则构建频繁模式智能发现系统,为网站拥有者改进站点架构建设提供更丰富的决策.

参考文献:

[1] B Berendt, A Hotho, G Stumme . Usage Mining for and on the Semantic Web[A]. In: Data Mining Next Generation Challenges and Future Directions[C]. Boston: AAAI Press, 2004. 461 – 481.

[2] N Stojanovic, J Gonzalez, L Stojanovic. ONTOLOGER: a system for usage-drivenmanagement of ontology-based information

portals[A]. Proceedings of the 2nd Internati-onal Conference on Knowledge Capture[C]. New York: ACM, 2003. 172 – 179.

[3] F A Lisi. Principles of inductive reasoning on the semant-ic web: a framework for learning in AL-log[A]. Proceedi-ngs of the 3rd International Workshop on Principles and Practice of Semantic Web Reasoning, LNCS 3703[C]. Heidelberg: Springer Berlin, 2005. 118 – 132.

[4] JJózefowska, A Lawrynowicz, T Lukaszewski. A study of the SEMINTEC approach to frequent pattern mining[A]. Proceedings of Prior Conceptual Knowledge in Machine Learning and Knowledge Discovery[C]. Warsaw, Poland: 2007. 41 – 52.

[5] F M Donini, M Lenzerini, D Nardi, et al. AL-log: inte-grating datalog and description logics[J]. Intelligent Info-rmation Systems, 1998, 10(3): 227 – 252.

[6] B Motik, U Sattler, R Studer. Query answering for OWL-DL with rules[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2005, 3(1): 41 – 60.

[7] B Motik, U Sattler. A comparison of reasoning techniques for querying large description logic ABoxes[A]. Proceed-ings of the 13th International Conference on Logic for Programming Artificial Intelligence and Reasoning[C]. Heidelberg: Springer Berlin, 2006. 227 – 241.

[8] M Sun, B Chen, M T Zhou. An ILP approach to mine the association rules on log ontology[A]. Proceedings of the IEEE International Conference on Apperceiving Compu- ting and Intelligence Analysis 2008[C]. Chengdu, China: IEEE Press, 2008. 274 – 278.

[9] U Hustadt, B Motik, U Sattler. Reducing SHIQ? descry-ption logic to disjunctive datalog programs[A]. Proceed-ings of the 9th Int. Conf. on the Principles of Knowledge Representation and Reasoning[C]. Whistler, Canada: 2004. 152 – 162.

[10] S Nijssen, J N Kok. Efficient frequent query discovery in Farmer[A]. Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, LNCS 2838[C]. Heidelberg: Springer Berlin, 2003. 350 – 362.

[11] J Józefowska, A Lawrynowicz, T Lukaszewski. On redu-cing redundancy in mining relational association rules from the Semantic Web[A]. Proceedings of the 2nd Int. Conf. on Web Reasoning and Rule Systems, LNCS 5341[C]. Karlsruhe: Springer, 2008. 205 – 213.

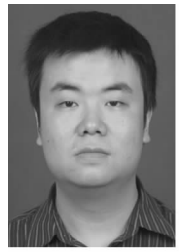
[12] F A Lisi, D Malerba. Inducing multi-level association rules from multiple relations[J]. Machine Learning, 2004, 55(2): 175 – 210.

[13] E Dantsin, T Eiter, G Gottlob, et al. Complexity and exp – ressive power of logic programming[J]. ACM Computing Surveys, 2001, 33(3): 374 – 425.

作者简介:

孙　明　男,1978 年 8 月出生于四川南充,现为电子科技大学计算机科学与工程学院博士生,主要研究领域为知识发现,语义 Web 使用挖掘以及本体学习.

E-mail: sunm@uestc.edu.cn



陈　波　男,1977 年 5 月出生于四川德阳,现为电子科技大学计算机科学与工程学院讲师、主要研究领域为知识发现以及粗糙集.

E-mail: bluesbeyond@vip.sina.com

周明天　男,1939 年 3 月出生于广西容县,中国电子学会会士、IEEE 高级会员,现为电子科技大学教授,博士生导师,主要研究领域为中间件技术,知识发现以及网络计算.