基于空间分割的SWRL数据集关联规则挖掘*)

袁 柳 李战怀 陈世亮

(西北工业大学计算机学院 西安 710072)

摘 要 语义Web环境下的关联规则挖掘是数据挖掘领域新的研究热点。本文针对SWRL数据集的特征,建立新的数据挖掘形式背景,将FCA用于关系型关联规则的挖掘,提出了基于搜索空间分割的关联规则挖掘方法。采用FCA作为频繁模式的压缩表示方式,从生成的闭查询导出的关联规则,可有效控制冗余规则的产生。将搜索空间进行划分可减小问题的规模,充分利用已有的挖掘过程的中间结果所提供的信息,减少了计算量。由于采用了分而治之的策略,本文的方法易于扩展到对海量语义Web数据的并行处理。

关键词 SWRL,FCA,关联规则

Search Space Partition-based Association Rules Mining from SWRL Data Set

YUAN Liu LI Zhan-huai CHEN Shi-liang

(School of Computer Science, Northwestern Polytechnical University, Xi'an 710072,China)

Abstract Association rules mining in Semantic Web is a new challenge for data mining researchers. A search space partition based association rules mining method for SWRL data set is proposed. FCA is adopted as the condensed representation of frequent patterns with conjunctive query formation, association rules induced from closed query can avoid producing redundant rules in semantic. Search space partition can divide the large scale problem into small parts, computation results obtained during mining procedure are also be used fully for reducing unnecessary computation. The method proposed can easily be adapted to parallel algorithm for processing large mount of Semantic Web Data because of the strategy of divide and conquer.

Keywords SWRL, FCA, Association rules

1 引言

近年来,数据挖掘领域有关事务型数据关联规则挖掘的研究已经较为成熟,提出了大量有效的关联规则挖掘算法。同时,对于一些更为复杂的数据类型,如树(tree)[2]、图(graph)[3]等结构上的关联规则挖掘研究也取得了一定的成果。然而,几乎所有的关联规则挖掘方法都仅适用于事务型数据库。语义Web的发展使得挖掘语义Web中的数据成为数据挖掘领域一个新的研究热点。与传统数据挖掘技术相比,语义Web挖掘最大的特点是它所处理的数据在形式上更复杂。语义Web规则语言SWRL(Semantic Web Rule Language)[1]在OWL的基础上增加了规则表示,因其具有强大的表达能力,使用日益广泛,因此从以SWRL表示的数据中挖掘信息具有一定的研究和应用价值。本文将关注SWRL数据集中关联规则的挖掘。与传统的关联规则挖掘过程类似,从SWRL数据集中挖掘关联规则也分为两个阶段:首先生成所有的频繁模式;其次根据频繁模式导出关联规则。海量数据是Web环境应用的特点,生成的频繁模式数量可能是巨大的,从而导致过量的关联规则,这不但增加计算负担,过多的关联规则也使得用户不易理解。传统的关联规则挖掘中,项集的压缩表示是防止产生冗余关联规则的重要方式,FCA(Formal Concept Analysis)是一种广泛采用的项集压缩表示的理论依据。本文将FCA的思想引入到SWRL数据集中的关联规则挖掘,并针对SWRL的数据特点,提出基于空间分割的SWRL数据集上的关联规则挖掘方法。

2 相关工作

与本文工作相关的研究主要有两方面:一是将FCA用于项集压缩表示的方法;二是以逻辑为基础的数据集关联规则的挖掘。在传统的事务型数据库中,已有不少将FCA用于关联规则挖掘的研究成果。Close[4]是第一个将FCA理论用于关联规则挖掘的算法。在此之后,研究者相继提出了A-Close[5], Pascal[6], Closet[7], Titanic[8]等一系列算法,每种算法都试图通过建立数据所蕴涵的闭包系统实现频繁项集的压缩表示,从而产生关联规则。目前基于描述逻辑(Description Logics,DL)知识库的复杂结构数据的关联规则挖掘的研究成果并不多。相关的研究主要集中在两方面:RDM(Relational Data Mining)和从强表达能力语言表示的数据中挖掘信息。典型代表分别是WARMR[9]和SPADA[10]。文献[12]以连接查询的形式表示频繁模式,针对OWL-DLP提出了支持可包含不同粒度级别的概念的频繁模式发现方法。但几乎所有的方法都没有考虑到频繁模式的等价性,因此会产生大量语义上冗余的关联规则。文献[11]首次将FCA用于Datalog数据的关联规则挖掘,提出了Datalog的冰山查询格结构,挖掘Datalog数据集中的关联规则。生成形式概念是使用FCA挖掘关联规则的前提,但到目前为止,以逻辑为基础的数据集上形式概念生成的研究成果仍然较少。

3 相关技术

语义Web可表示成一种多层的体系结构,OWL作为本体层的语言,以描述逻辑DL为基础。OWL的一个明显不足是,其支持的关于属性的操作相当贫乏。例如,它不支持属性的合成构造运算,OWL中不能表达"如果y是x的父亲,z是y的兄弟,则z是x的叔叔"的语义,但这在Horn逻辑中通过规则很容易实现。在语义Web的逻辑层,通过建立规则的方式对OWL进行扩展,弥补其表达能力上的缺陷,SWRL正是

*)国家自然科学基金项目资助(60573096)。袁 柳 博士生,主要研究方向为语义Web及信息检索;李战怀 博士,教授,博士生导师,主要研究方向为数据库理论与技术;陈世亮 博士生,主要研究方向为多媒体信息管理、多媒体信息检索。

这种方式的一种实现。SWRL 的基本思想是在 OWL 基础上通过增加规则的描述，在增强表达能力的同时最大限度地向后兼容 OWL 现有的语法和语义。SWRL 向 OWL 中增加了一种新的公理形式——Horn 子句规则，可以对 OWL DL 抽象语法和直接模型论语义进行扩展，从而提供一种包含 Horn 规则的 OWL 本体的正规语义和语法。SWRL 数据集具有 DL 知识库的结构，包含 Tbox 和 Abox 两部分。不同的是，Tbox 中除了关于概念的断言，还包括了若干规则，这些规则所表达的关系不能用 DL 提供的构词来描述。规则以 Horn 子句的形式描述：$H \leftarrow B_1, \cdots, B_n$。其中 $H, B_1, \cdots, B_n$ 均为原子，$B_1, \cdots, B_n$ 为 Tbox 中已有的概念或关系名称，H 为由规则定义的不在已有 Tbox 中的概念或关系名称。

以下对本文中用到的逻辑术语进行简要的说明。项(term)定义为常量或变量，原子(atom)为一个 m 元的谓词符号。确定子句(definite clause)为具有如下形式的全称量词约束公式：$B \leftarrow A_1, \cdots, A_n (n \geqslant 0)$，其中 B 和 A_i 均为原子。当 $n=0$，该子句被称为事实。替换(substitution)θ 为一个集合 $\{X_1/a_1, \cdots, X_m/a_m\}$，将变量 X_i 与项 a_i 绑定，即将子句中的 X_i 用 a_i 替换。子句 c 关于替换 θ 的实例可以通过将 c 中出现的变量 X_i 分别以项 a_i 替换而得到。若 $C\theta$ 中仅包含常量项，不含变量，则 $C\theta$ 称为 C 的基例，θ 称为基替换。

4 SWRL 数据集中关联规则的表示

在 SWRL 数据集中挖掘关联规则与传统的关联规则挖掘过程类似，首先产生频繁模式，然后在频繁模式的基础上导出关联规则。传统关联规则挖掘中通常以项集表示模式，而在本文的数据挖掘环境中，模式具有不同的形式。

定义 1 SWRL 数据集上的模式具有如下形式：

$$Q = q(X) \leftarrow C_{key}(X), \alpha_1, \cdots, \alpha_n$$

其中 C_{key} 是一个一元谓词，它是出现在 SWRL 数据集中的任意概念，表示用户在频繁模式发现中所关注的概念。X 也是 Q 中唯一的受 Ckey 约束的变量，$\alpha_i (1 \leqslant i < n)$，是数据集中出现的概念(一元谓词)或属性(二元谓词)。除变量 X 之外，α_i 还可约束其他变量。该形式可以简单表示为

$$Q = C_{key}, \alpha_1, \cdots, \alpha_n$$

这种形式的模式也可以看作连接查询，表示查询同时满足谓词 C_{key} 和 α_i 的个体。本文中提及的查询均具有定义 1 的形式。根据数据集可产生的所有具有该形式的模式构成了候选频繁模式的搜索空间 L。关于候选频繁模式的产生规则不是本文讨论的主要问题，在此不赘述。

数据集 r 中，具有定义 1 形式的连接查询 Q 的回答是满足 Q 的基替换的集合，即使得 $Q\theta$ 为真的基替换 θ 的集合，记为 answerset(Q, r)。指定关键概念 C_{Key}，设 $\hat{Q} = C_{Key}(X)$，则 Q 的支持度可定义为

$$\text{supp}(Q, r, C_{key}) = \frac{|\theta \in answerset(\hat{Q}, r) \mid Q\theta \text{为真}|}{|\theta \in answerset(Q, r)|}$$

例 1：假设数据集中包含以下事实：

customer(*allen*) parent(*allen*, *bill*) buys (*allen*, *wine*)

customer(*bill*) parent(*allen*, *carol*) buys (*bill*, *cola*)

customer(*carol*) parent(*bill*, *zoe*) buys (*bill*, *pizza*)

customer(*diana*) parent(*carol*, *diana*) buys (*diana*, *pizza*)

以及如下规则

grandparent(X,Z) ← parent(X, Y), parent(Y,Z)

考虑查询 $Q = q(X)$(customer(X), parent(X, Y), buys (Y, pizza)，该查询的回答 answerset(Q, r)是一个包含了两个基替换的集合：{(X/allen, Y/bill), (X/carol,Y/diana}。

从具有定义 1 形式的频繁模式中，可导出关联规则，称这种形式的关联规则为关系型关联规则(Relational Association Rules)。

定义 2 关系型关联规则是具有如下形式的规则：

$$l_1, \cdots, l_m \rightarrow l_{m+1}, \cdots, l_n$$

l_i 为数据集中出现的谓词，$1 \leqslant i < n, 1 \leqslant m < n$，查询 $Q_b = l_1, \cdots, l_m$ 为规则体，子查询 $Q_h = l_{m+1}, \cdots, l_n$ 为规则头，$Q_c = l_1, \cdots, l_m, l_{m+1}, \cdots, l_n$ 为规则的结论。

注意到，关系型关联规则由两个查询组成，第二个查询可作为第一个的扩展。由此可定义关系型关联规则 Rar 的支持度 supp(Rar)和置信度 conf(Rar)：

$$\text{supp}(Rar) = \text{supp}(Q_c, r, C_{key})$$

$$\text{conf}(Rar) = \frac{\text{supp}(Q_c, r, C_{key})}{\text{supp}(Q_b, r, C_{key})}$$

例 2：对于例 1 中的数据集，可产生如下规则(括号中数据分别为规则的支持度和置信度)：

R1：customer(X)，buys(X, *wine*) →parent(X, Y)，parent(Y, Z) (0.25，1/1)；

R2：customer(X)，parent(X, Y) →buys(Y, *cola*) (0.25，1/3)；

R3：customer(X)，parent(X, Y) →buys(Y, *cola*)，buys(Y, pizza) (0.25，1/3)。

在事务型数据集上，如果两个项集同时被同一事务集合所包含，可认为这两个项集是等价的，等价的项集集合称为一个等价类。该思想同样可用于 SWRL 数据集上，将相互等价的查询定义为一个等价类。观察例 2 中的关联规则，$R3$ 包含了 $R2$ 且 $R3$ 和 $R2$ 描述了相同的实例数据。在给定的数据集上，一些规则包含了完全相同的实例数据，因此具有完全相同的支持度和置信度。在这种情况下，可以删除那些更特殊(more specific)的规则，而不会丢失任何信息。

5 基于空间分割的 SWRL 关联规则挖掘

5.1 频繁模式的压缩表示

压缩表示的核心问题是，如何从等价类中找到其中的一个模式代表该类中的其他模式，而并非找到所有的模式。与传统的模式等价性判定相比，SWRL 模式等价性的判定要复杂得多。首先，计算所有的频繁模式需要很高的计算代价，通常采用的判断模式间包含关系的运算 θ-subsumption 具有较高的时间复杂度，并且模式频繁度的计算也是一项费时的工作。其次，在频繁模式发现过程中，需要产生、测试大量的候选模式。与简单的模式语言(如项集)相比，以连接查询构成的模式空间巨大。ILP(Inductive Logic Programming)方法是从子句模式空间中发现频繁模式的主要方式，目前已有的方法一般都遵循的原则是子句必须满足一定的语法约束(也称为语言偏置，language bias)，并且在求精规则的导引下，产生所有的候选频繁模式。

FCA(Formal Concept Analysis)[11]理论提供了频繁项集的压缩表示机制。本文拟将其用于 SWRL 数据集中的频繁模式表示。在形式概念分析中，形式背景通常定义为一个三元组 $K = (G, M, I)$，其中 G 是对象集合，M 是属性集合，$I \subseteq G \times M$ 是 G 与 M 之间的一个二元关系。若 $(g, m) \in I$，表示“对象 g 具有属性 m”。

对 $A \subseteq G$，定义 $A' = \{m \in M \mid \forall g \in A, (g, m) \in I\}$，

对 $B \subseteq M$，定义 $B' = \{g \in G \mid \forall m \in B, (g, m) \in I\}$

形式背景 K 上的一个概念定义为一个二元组 (A, B)，满

足 $A\subseteq G, B\subseteq M, A'=B, B'=A$。其中 A 称为概念(A,B)的外延，B 称为概念(A,B)的内涵。形式背景 K 上的任何两个概念(A,B)和(C,D)，如果 $B\subseteq D$，则称(A,B)是(C,D)的超概念，(C,D)称为(A,B)的子概念，记为$(C,D)\leqslant(A,B)$。设 $B(K)$为形式背景 K 上的所有形式概念集合，则该集合中的元素可形成一个有序集 $B(K)=(B(K),\leqslant)$。上文所述的等价关系 ψ 定义如下：

$X,Y\in B(M)$，$X\psi Y$ 当且仅当 $X'=Y'$，$B(M)$表示 M 的幂集。

可以证明每个概念的内涵恰好是它所在等价类中的最大项集，称概念内涵为闭项集。定义 $B(M)$上的闭包运算符为 $B\to B''$。对任意项集 $B\in B(M)$，其闭包为 B''，正是 B 所在等价类的概念内涵。对任意 $\hat{B}\in B(M)$，且 $B\psi\hat{B}$，有 $\mathrm{supp}(\hat{B})=\mathrm{supp}(B)=\mathrm{supp}(B'')$。形式背景 K 上的形式概念(A,B)的支持度定义为项集 B 的支持度 S_B（$S_B=\frac{|A|}{|G|}$）。支持度大于预定义的支持度阈值的概念称为频繁概念，所有频繁概念构成的集合可构成冰山概念格(iceberg concept lattice)[13]。

5.2 将 FCA 用于 SWRL 数据集关联规则挖掘

由于模式表示方式的差异，传统形式背景 K 的定义不适合直接用于本文的数据挖掘环境。下面给出一种新的形式背景的定义方式。

定义 3 设 r 表示 SWRL 数据集，L 为包含关键概念 C_{key} 的连接查询的集合，集合中每个查询 Q 都包含关键谓词 C_{key}。关于 r，L，C_{key} 的形式背景可定义为

$$K_{r,L,C_{\mathrm{key}}}:=(G_{r,L,C_{\mathrm{key}}},M_{r,L,C_{\mathrm{key}}},I_{r,L,C_{\mathrm{key}}})$$

其中 $G_{r,L,C_{\mathrm{key}}}=\mathrm{answerset}(\hat{Q}=C_{\mathrm{key}}(X),r)$，

$M_{r,L,C_{\mathrm{key}}}=L$，$(\theta,Q)\in I_{r,L,C_{\mathrm{key}}}$ 当且仅当存在一个 Q 的一个基替换 $\bar{\theta}$ 使得 $\theta\subseteq\bar{\theta}$。

关于定义 3，需要注意以下几点：

(1) 在事务型数据库中，项集的组合(如项集的并)可能不再是一个项集，而查询在连接运算下仍是一个查询，设 q_1，q_2 为两个查询，则 $q_1\wedge q_2$ 仍为一个查询。这意味着查询集合 L 对于“$\wedge$”操作运算是封闭的；

(2) 通常不考虑查询的集合而仅考虑单个查询的性质，如查询的等价性。

$K_{r,L,C_{\mathrm{key}}}$ 中的形式概念可以采用通用的形式概念生成方法实现。根据以上的定义，规定 $K_{r,L,C_{\mathrm{key}}}$ 中的形式概念(A,B)的内涵为连接查询 $\bigwedge_{i=1}^{n}B$(即 B 包含的所有查询做连接运算，假设 B 中有 n 个查询)，表示 B 中所有查询的合取。这样的查询也称为关于 $K_{r,L,C_{\mathrm{key}}}$ 的闭查询。两个查询 SQ_1、SQ_2 是关于数据集 r 的等价查询当且仅当 $\mathrm{answerset}(SQ_1,r)=\mathrm{answerset}(SQ_2,r)$，记为 $SQ_1\Psi_{r,L,C_{\mathrm{key}}}SQ_2$，表示 SQ_1、SQ_2 是等价的。每个等价类中的最特化(most specific)查询正是其所对应的关于形式背景 $K_{r,L,C_{\mathrm{key}}}$ 的形式概念的闭查询。为了挖掘关系型关联规则，我们希望在 $B(M_{r,L,C_{\mathrm{key}}})$上找出所有的闭集。

5.3 基于空间分割的 FCA

基于概念格的关联规则挖掘主要包括以下三步：

(1)找出所有频繁概念的内涵(频繁闭项集)；

(2)从第 1 步得到的频繁内涵生成所有频繁项集；

(3)从第 2 步生成的每个频繁项目集 I，产生所有置信度大于等于 minconf 的关联规则。

其中第 1 步是问题的核心，提供了随后步骤所需的所有信息。实际应用中形式背景是巨大的，尤其对 Web 环境下的数据，对此提出空间分割的方法实现形式概念的生成。

对模式空间进行有效的分割，借鉴文献[14]的方法，首先给出一些必要的定义。

设出现在数据集中的所有谓词集合为 D，在 D 上定义一个线性序列 $D=\{d_1<d_2<\cdots<d_{|D|}\}$，搜索空间 $SQ=L$。同时，也可定义 SQ 上的线性序列 $SQ=\{q_1<q_2<\cdots<q_{|P|}\}$。对 $Q\subseteq SQ$，称 $q\in Q$ 是 Q 的最大元素，表示为 $\max(Q)=q$，如果 $\forall q_i\in Q$ 且 $q_i\neq q$ 有 $q_i<q$。特别地，如果 $Q=\{q\}$，则 $\max(Q)=q$。对 $Q\subseteq SQ$，称 Q' 是正规的，如果 $\forall q_i\in Q'\backslash Q$，有 $\max(Q)<q_i$。

每个查询 $q_i\in SQ$ 以向量形式表示。对于向量所表示的查询，若分量所表示的谓词在查询中出现，则该分量值为 1，否则为 0。以例 1 为例，查询 $q(X)\leftarrow(\mathrm{customer}(X),\mathrm{Parent}(X,Y),\mathrm{buy}(X,cola)$可表示为$(1,1,1)$。本文假设 SQ 中的所有查询都包含用户指定的关键概念 C_{key}。根据查询所包含的谓词，可以将搜索空间 SQ 划分为若干个子空间。以下给出通过谓词定义搜索空间子空间的方法。

定义 4 设 $S(\mathrm{key},\mathrm{Cand})=\{d\mid d=\mathrm{key}\cup c,c\in B(\mathrm{Cand})\}$ 为谓词向量中所包含的谓词集合，其中，$\mathrm{key}\subseteq D$，$\mathrm{Cand}\subseteq D$，且 $\forall d_i\in\mathrm{Cand}$，$\max(\mathrm{key})<d_i$。$B(\mathrm{Cand})$ 表示 Cand 的幂集。Key 表示谓词向量必须包含的谓词集合，Cand 是可能出现的谓词集合，称 $S(\mathrm{key},\mathrm{Cand})$所能表示的查询为 SQ 的一个搜索空间。

根据如上定义和预先定义的 SWRL 中谓词的线性序，可以将形式背景 $K_{r,L,C_{\mathrm{key}}}$ 确定的搜索空间用特征向量的形式表示如表 1，表中第一列为对谓词向量所设定的编号。

表 1 搜索空间 S(key, Cand)

	d_n	d_{n-1}	d_{n-2}	…	d_3	d_2	d_1	C_{key}
1	0	0	0	…	0	0	0	1
2	0	0	0	…	0	0	1	1
3	0	0	0	…	0	1	0	1
4	0	0	0	…	1	0	0	1
				…				1
$2^{\lvert dn\rvert}-1$	1	1	1	…	1	1	0	1
$2^{\lvert dn\rvert}$	1	1	1	…	1	1	1	1

定义 5 给定搜索空间 $S(\mathrm{key},\mathrm{Cand})$，称 $\mathrm{SubS}_i(\mathrm{key}_i,\mathrm{Cand}_i)=\{d\mid d=\mathrm{key}_i\cup c,c\in B(\mathrm{Cand}_i)\}$ 为由谓词 d_i($d_i\in\mathrm{Cand}$)确定的子搜索空间。其中 $\mathrm{key}_i=\mathrm{key}\cup\{d_i\}$，$\mathrm{Cand}_i=\{d\mid d_i<d,d\in\mathrm{Cand}\}$。

根据定义 5，可以将给定搜索空间 $S(\mathrm{key},\mathrm{Cand})$划分为 $|\mathrm{Cand}|+1$ 个子搜索空间。对表 1 给出的搜索空间，表 2 给出分别由每个谓词确定的子搜索空间 S_i，$i=0,1,\cdots,|D|$。

由于每个谓词可以约束多个变量，一个向量可以表示多个查询。例如，假设数据集中含有四个不同的谓词，分别是概念谓词 C_{key} 和 C_1，属性谓词 P_1 和 P_2，可表示为$(C_{\mathrm{key}},C_1,P_1,P_2)$，则查询 Q_1：$q(X)\leftarrow C_{\mathrm{key}}(X),P_1(X,Y),C_1(X)$ 与查询 Q_2：$q(X)\leftarrow C_{\mathrm{key}}(X),P_1(X,Y),C_1(Y)$ 具有相同的向量表示形式$(1,1,1,0)$。因此在每个子搜索空间中，每行向量都对应着一个查询的集合。

对于分割的搜索空间，有如下性质：

性质 1 给定搜索空间 $S(\mathrm{key},\mathrm{Cand})$，对 $\forall i\neq j$，如果 $S_i(\mathrm{key}_i,\mathrm{Cand}_i)$和 $S_j(\mathrm{key}_j,\mathrm{Cand}_j)$为 S 的子搜索空间，则有 $S_i\cap S_j=\varnothing$，且 $\bigcup_{0}^{|\mathrm{Cand}|}S_i=S$。

性质 2 给定搜索空间 $S(\mathrm{key},\mathrm{Cand})$，$S_i(\mathrm{key}_i,\mathrm{Cand}_i)$为

谓词 d_i 确定的子搜索空间，对 $\forall s \in S_i$，如果 s'' 是正规的，则 $s'' \in S_i$。

性质 1、性质 2 可根据定义 4、定义 5 得到。由这两个性质可以得到如下结论：将一个搜索空间划分为若干个子搜索空间，在每个子搜索空间搜索正规闭集，并且能够保证在所有子搜索空间找到的正规闭集的并等于在原搜索空间直接搜索到的闭集的集合。

表 2　子搜索空间 S_i

	d_n	d_{n-1}	d_{n-2}	…	d_3	d_2	d_1	C_{key}
S_0	0	0	0	0	0	0	0	1
S_1	0	0	0	0	0	0	1	1
S_2	0	0	0	0	0	1	0	1
	0	0	0	0	0	1	1	1
S_3	0	0	0	0	1	0	0	1
	0	0	0	0	1	0	1	1
	0	0	0	0	1	1	0	1
	0	0	0	0	1	1	1	1
…				…				1
	1	0	0	0	0	0	0	1
	1	0	0	0	0	0	1	1
$S_{\|D\|}$				…				1
	1	1	1	1	1	1	0	1
	1	1	1	1	1	1	1	1

5.4　子空间中频繁概念的搜索

将一个搜索空间划分为若干子搜索空间后，并不是在所有的子搜索空间都能找到正规的闭集。对于不能生成正规闭集的子搜索空间可以不必考虑，这样可以有效提高算法的效率[14]。但仅考虑这一特征对于 SWRL 数据集中是不够的。挖掘关系型关联规则，首先要找到频繁闭项集，因此必须考虑概念的支持度。子空间 S_i 中，在形式背景（$G_{r,L,C_{key}}$，$M_{r,L,C_{key}}$，$I_{r,L,C_{key}}$）下的形式概念(A, B)若满足 $\frac{|A|}{|G_{r,L,C_{key}}|} \geqslant$ minsupp，则为频繁概念。然而并不是所有的子空间都存在频繁概念。称不能找到频繁概念的子空间是无效的。在传统的频繁项集发现中，有一条重要的性质，即非频繁项集的超集一定是非频繁项集。将其用于本文的数据挖掘环境中可得以下结论：

引理　对于具有定义 1 形式的查询的集合 Q，对 $\forall q_i, q_j \in Q$，若 q_i 或 q_j 为非频繁查询，则查询 $q_i \wedge q_j$ 一定是非频繁的。

证明：根据 q 支持度定义，只有当使 $q\theta$ 为真的基替换个数达到一定数量时，q 才是频繁的。若 q_i 或 q_j 为非频繁查询，那么同时满足 q_i 和 q_j 的基替换必定不能满足频繁度阈值的要求，因此 $q_i \wedge q_j$ 一定是非频繁的。 □

定理　如果 $C_{key}(X) \wedge d_i(X)$ 是非频繁的，则 d_i 确定的子空间一定是无效的。

证明：因为 $C_{key}(X) \wedge d_i(X)$ 是非频繁的，由 d_i 确定的子空间中的每个查询，至少都包含了谓词 C_{key} 和 d_i。根据引理，该子空间中的每个查询都是非频繁的。显然该子空间中不可能产生频繁概念，因此该子空间无效。 □

为了方便判定子空间的有效性，对每个谓词 d_i，预先计算查询 $C_{key}(X) \wedge d_i(X)$（若 d_i 为概念谓词）或查询 $C_{key}(X) \wedge d_i(X, Y)$（或 $C_{key}(X) \wedge d_i(Y, X)$，若 d_i 为属性谓词）的支持度。注意到，无论 d_i 为概念还是属性，$C_{key} \wedge d_i$ 为 d_i 确定的子空间中的第一个向量所表示的查询。

排除了无效子空间，对其余的子空间可进一步分割，仅在可能产生频繁概念的子空间中搜索频繁概念。由于对谓词进行了排序，在判断子空间 S_i 的有效性时，可充分利用 S_j 有效性判断的计算结果。

例 3：对于例 1 的数据集，以 customer 为关键概念 C_{key}，得到的子空间分割结果如表 3 所示。

表 3　子空间分割结果

	parent	buy	customer
S_0	0	0	1
S_1	0	1	1
S_2	1	0	1
	1	1	1

若确定支持度阈值为 0.25，根据定理不能够排除无效的子空间，因此对每个子空间分别进行考察。子空间 S_0 中可得到闭查询 Q_1 = customer(X)，对产生关联规则是无意义的；子空间 S_1 中可得到闭查询 Q_2 = customer(X)，buys(X，pizza)；子空间 S_2 中包含两个谓词向量，(1，0，1) 对应闭查询 Q_3 = customer(X)，parent(X，Y)；(1，1，1) 对应如下闭查询：

Q_4 = customer(X)，parent(X，Y)，buys(Y，pizza)；

Q_5 = customer(X)，parent(X，Y)，buys(X，cola)，buys(X，pizza)；

Q_6 = customer(X)，parent(X，Y)，parent(X，U)，buys(Y，pizza)，parent(Y，Z)，parent(U，V)，buys(X，wine)，buys(Y，cola)，buys(V，pizza)。

目前已有多种利用 FCA 提取关联规则的方法，且不损失信息。这些方法都可用于本文 SWRL 挖掘的形式背景来生成关系型关联规则。根据得到的闭查询，可得到如下规则（括号内为规则的置信度）：

从 Q_2 得到：customer(X)→buys(X，pizza)　(1/2)

从 Q_3 得到：customer(X)→parent(X，Y)　(3/4)

从 Q_4 得到：customer(X)，parent(X，Y)→buys(Y，pizza)　(2/3)

从 Q_5 得到：customer(X)，parent(X，Y)→buys(X，cola)　(1/3)

从 Q_5 得到：customer(X)，buys(X，pizza)→buys(X，cola)　(1/2)

从 Q_6 得到：customer(Y)，parent(X，Y)，buys(Y，pizza)→parent(Y，Z)，parent(U，V)，buys(X，wine)，buys(Y，cola)，buys(V，pizza)　(1/2)

5.5　算法描述和性能分析

以下是本文算法框架的描述。其中在谓词集合 D 上已定义好一个线性序列，Core 为搜索空间必须包含的概念集合，初始情况下 Core=$\{C_{key}\}$。

算法：SearchSpacePartition(D, $Core$)

```
1 for each d_i in D
  {
2     if Core∪d_i 表示的查询均为非频繁
3     标记 d_i 确定的子空间无效；
4 for each d_j in D 且 j>i
  //对有效子空间进一步分割
5     SearchSpacePartition(D,Core∪{d_j})；
```

6 计算子空间上的闭查询；
7 从闭查询中导出关联规则；
}

本文使用 Java 语言，以 KAON2 API[15]作为处理 SWRL 的工具，对所提出的方法进行验证。KANO2 API 主要用于构造包含关键概念的查询集合，以及对每个查询计算其基替换。本文又选择了 STULONG[16] 的研究数据进行测试。STULONG 是一个关于动脉硬化风险因素的纵向研究项目，在先前的研究工作中，我们已建立了 STULONG 研究数据的 SWRL 表示形式。设定模式中除 C_{key} 最多包含 4 个谓词，对该数据集的测试证明了方法的正确性和有效性。表 4 以生成频繁模式的数量和时间说明了方法的有效性。$C1$ 为采用本文方法的结果，$C2$ 为采用传统的基于简单求精规则的频繁模式生成方法。可以看到，采用本文方法直接可得到闭查询，数量较少，而一般的方法直接产生的频繁查询，其中包含等价查询，数量较多。从运行时间上看，本文方法也占优势。但实验还发现，目前的实现若要处理大量数据，还需要对算法进行更深入的优化以提高效率。

表 4 实验结果

查询长度	查询数量		运行时间	
	$C1$（频繁闭查询）	$C2$（频繁查询）	$C1$	$C2$
1	7	7	43s	54s
2	21	31	89s	107s
3	59	152	262s	317s
4	92	628	1987s	2122s

本文算法效率主要受两方面因素的影响：形式概念的生成和子空间有效性的判定。形式概念生成算法的选择需要考虑数据集的特征。例如，对于大而稠密的形式背景基于闭包运算的概念生成算法的性能最好，小而稀疏的形式背景适宜采用渐近式算法。由于真实的 Web 数据一般都是海量的，因此适合选择基于闭包的算法。但这类算法存在的问题是，为了避免概念的重复生成，算法往往采取一些策略，如以某种规定的顺序生成所有的闭包。然而当所生成的闭包不满足规定的顺序时，则该闭包无效，这种计算无效闭包的过程是对资源的浪费。空间分割的思想可解决这个问题，将闭包空间视为搜索空间并进行划分，只对有效的子空间进行搜索生成概念，这样可降低计算量。为了计算子空间的有效性，需要计算并保存一些谓词的频繁度计数作为中间结果，查询频繁度的计算不但耗时，还需要额外的空间消耗。考虑形如定义 3 的 SWRL 形式背景，G 为所有满足查询 $Q=C_{key}(X)$ 的替换，M 为所有包含关键谓词 C_{key} 的查询，G 和 M 不但涉及到出现在数据集中的谓词，还必须明确谓词所约束的变量，因此比传统事务型数据集上的形式背景表示复杂。对长度为 L_Q 的查询，至少需要额外的 L_Q 个空间存储谓词所约束的变量。

在传统的事务型数据集上，设最大频繁封闭项集的大小为 L，数据库对象个数为 O，项目个数为 I，候选频繁项集个数的最坏情况为 N，对数据库的访问时间为 db，则采用 A-Close 算法的时间复杂度为 $O(L(db+N\times O\times I))$。将 A-Close 应用到定义 3 的形式背景中，则 L 为最大的频繁闭查询长度（即频繁查询中包含的谓词个数），O 为满足查询 $Q(X)=$key(X) 的替换个数，I 为数据集中出现的谓词个数，N 为最坏情况下候选频繁模式的个数。不同于传统情况下，通过遍历数据库即可判定 $(g,m)\in I$ 是否成立，SWRL 数据集中该关系的判定需要推理的支持。众所周知，描述逻辑上的推理是一个及其耗时的过程，例如 SHIQ 概念相对于概念层次和传递角色的可满足性推理是 EXPTIME 完备问题。为了减少推理过程对概念生成的影响，可以预先计算部分需要推理的中间结果。

结束语 本文通过分割模式空间的方式，挖掘 SWRL 数据集中的关联规则，并将 FCA 应用于挖掘过程。首先建立适合于 SWRL 数据的形式背景，利用空间分割的策略找到子空间的闭查询，关联规则仅从闭查询中导出，可有效防止冗余关联规则的产生。目前还没有一个通用标准的数据集用来检测对 SWRL 数据的处理能力，我们目前仅使用特定的数据集对所提出的方法进行了说明。下一步工作主要在于对算法性能的改进和优化，在定义 3 的形式背景下，如何有效地生成形式概念，是影响本文方法效率的重要因素，是值得深入研究的问题。在实际应用背景的支持下，定制更有效的 SWRL 关联规则挖掘方法，并将其用于语义 Web 查询等应用中，也是下一步的工作目标。

参考文献

[1] http://www.w3.org/Submission/SWRL/
[2] Zaki M. Efficiently Mining Frequent Trees in a Forest// Proceedings of the Eight ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton, Alberta, Canada, 2002: 68-75
[3] Kuramochi M, Karypis G. Frequent Subgraph Discovery//Proceedings of the 2001 IEEE International Conference on Data Mining. San Jose, California, USA, 2001: 313-320
[4] Pasquier N, Bastide Y, Taouil R, et al. Efficient Mining of Association Rules Using Closed Itemset Lattices. Journal of Information Systems, 1999, 24(1): 25-46
[5] Pasquier N, Bastide Y, Taouil R, et al. Discovering Frequent Closed Itemsets for Association Rules//Proceedings of the 7th International Conference on Database Theory. Jerusalem, Israel, 1999: 398-416
[6] Bastide Y, Taouil R, Pasquier N, et al. Mining Frequent Patterns with Counting Inference. SIGKDD Explorations, Special Issue on Scalable Algorithms, 2000, 2(2): 66-75
[7] Pei J, Han J, Mao R. CLOSET: An efficient algorithm for mining frequent closed itemsets//The Proceeding of ACMSIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. 2000: 21-30
[8] Stumme G, Taouil R, Bastide Y, et al. Computing Iceberg Concept Lattices with Titanic. Knowledge and Data Engineering (KDE), 2002, 42(2): 189-222
[9] Dehaspe L, Toivonen H. Discovery of Frequent Datalog Patterns. Data Mining and Knowledge Discovery, 1999, 3(1): 7-36
[10] Lisi F A, Malerba D. Inducing Multi-level Association Rules from Multiple Relation. Machine Learning Journal, 2004, 55: 175-210
[11] Ganter B, Wille R. Formal Concept Analysis: Mathematical Foundations. Berlin Heidelberg: Springer-Verlag, 1999
[12] Józefowska J, Lawrynowicz A, Lukaszewski T. Towards discovery of frequent patterns in description logics with rules// The Proc of the International Conference on Rules and Rule Markup. Ireland, 2005: 84-97
[13] Stumme G. Iceberg Query Lattices for Datalog//The Proceeding of ICCS 2004. Kraków, Poland, 2004: 109-125
[14] 齐红，刘大有，等. 基于搜索空间划分的概念生成算法. 软件学报，2005，12：2029-2035
[15] http://kaon2.semanticweb.org
[16] http://euromise.vse.cz/challenge/index.html