模式级链接关联数据集上的关联规则挖掘研究

袁 柳[1] 张龙波[2]
(陕西师范大学计算机科学学院 西安 710062)[1] (山东理工大学计算机学院 淄博 255049)[2]

摘 要 针对关联数据集合呈现出的大数据特性和蕴含的语义信息,提出了首先建立关联数据集的模式级链接,再进行关联规则挖掘的方法。在同领域RDF数据集上定义RDF数据项模式并提出数据项模式的产生规则;利用RDF数据查询技术从数据项模式获得RDF数据项集合,进而再推导出特定领域内的关联规则。提出的基于关联数据RDF数据项模式的关联规则挖掘方法将关联规则挖掘扩展到同一领域内的数据集合而不再局限于单一数据集,同时给出了基于Hadoop的大规模RDF数据集上的关联规则挖掘的实现方案。实验结果验证了模式级链接对于关联规则挖掘的价值和所提方法的有效性。

关键词 语义大数据,关联数据,本体,RDF,关联规则
中图法分类号 TP311.1 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.9.017

Association Rules Mining on Schema-level Interconnected Associated Data

YUAN Liu[1] ZHANG Long-bo[2]
(College of Computer Science, Shaanxi Normal University, Xi'an 710062, China)[1]
(School of Computer Science and Technology, Shandong University of Technology, Zibo 255049, China)[2]

Abstract A schema-level interconnected association rules mining method for large scale associated data was proposed based on the semantic information implied in the associated data set. Instead of mining association rules from separated RDF data sets directly, firstly, we established schema-level linkage between different data sets. The RDF data item pattern generation rules are defined based on the schema-level linked datasets and then the RDF data query techniques are exploited for constructing RDF data items sets. The proposed data item patterns generation rules can extend the data mining objects from a single data set to multi-datasets in the same domain. A Hadoop based implementation plan of association rules mining was designed. The experiment results prove the value of establishing schema-level linkage on linked data and the effectiveness of the proposed method.

Keywords Semantic big data, Associated data, Ontology, RDF, Association rules

1 引言

随着开放关联数据(Linked Open Data, LOD)项目规模的不断壮大[1],越来越多的数据以遵循关联数据规范的形式在Web上发布,LOD所呈现出的大数据特性受到了数据管理研究领域的广泛关注,LOD也成为语义大数据的典型代表。来自LODStats(http://stats.lod2.eu)的统计数据显示,截至2015年7月,可访问的LOD数据集的总量已超过9960个,相比2015年年初1048个LOD数据集的规模,LOD的发展速度可见一斑。关联数据集采用资源描述框架(Resource Description Framework, RDF)三元组数据模型,与传统的事务型数据相比,RDF结构蕴含了更为丰富的领域背景知识并支持语义推理。如何从海量关联数据集中挖掘出有价值的信息是数据挖掘研究面临的一个新的挑战,目前该领域研究成果主要集中在对经典数据挖掘方法的改造上,针对LOD结构和语义特征的较为系统的研究成果还较为少见。

与经典的数据挖掘任务相比,关联数据集上的数据挖掘面临着两大困难:首先是数据集的选择,LOD将各种不同规模的RDF数据集使用owl:sameAs、rdf:type等谓词链接起来,从而构成了一个庞大的开放关联数据网络,对整个LOD中的RDF数据进行挖掘显然是不现实的,也是无意义的,如何从中选择合适的数据集合以满足数据挖掘任务是首先要解决的问题;其次,LOD中每个RDF数据集合可能使用不同的本体描述数据,因此会产生相同的数据具有不同的描述或相同的描述却对应着不同的数据的情况,这种不一致的数据描述会严重影响数据挖掘结果的可靠性。本文研究将关注后者,重点对相同领域内多个LOD数据集上的关联规则挖掘策略和方法展开深入研究。

2 相关工作

由于LOD被认为是"一种最可行的语义Web的实现方式",因此在理论上语义Web上的数据挖掘成果可用于LOD

到稿日期:2015-08-01 返修日期:2015-09-21 本文受国家自然科学基金项目:云计算环境下旅游信息个性化服务模型研究(41271387),中央高校基本科研业务费专项资金:模式级链接开放关联数据集上的数据挖掘关键技术研究(GK201503066)资助。
袁 柳(1979-),女,博士,讲师,主要研究方向为Web数据管理、语义信息检索,E-mail:yuanliu@snnu.edu.cn;张龙波(1968-),男,博士,教授,主要研究方向为数据流与数据挖掘。

上的挖掘[2,3]。大多数语义 Web 数据挖掘的实现都基于归纳逻辑编程(Inductive Logic Programing,ILP),ILP 可以充分地利用蕴含在语义 Web 数据底层的概念逻辑关系,实现深层次的推理与挖掘,但是需要将被挖掘数据重写成逻辑编程的格式,并且在发现有意义模式的能力上还有较大的不足。显然,基于 ILP 的方法并不适合于 LOD 的大数据特性。针对 RDF 三元组数据集模型在结构上的特征,有研究将用户所感兴趣的概念和在挖掘过程中需要考虑的属性定义为数据挖掘模式,并定义了语义 Web 数据集上项集、事务等关联规则挖掘过程中的核心概念,同时给出了生成项集、事务集合的具体算法,充分利用了领域本体中定义的概念及属性间的关系,从而将 RDF 数据集转化为经典 Apriori 算法可以处理的形式[4]。但是该研究的对象是单个 RDF 数据集,没有考虑到 LOD 中 RDF 数据集间的链接关系,也没有进一步考虑海量 RDF 数据的处理效率问题。有研究开始考虑如何对经典的数据挖掘方法进行改造以实现 LOD 挖掘[5],提出的语义 Web 关联规则挖掘算法 SWApriori 直接作用在 RDF 三元组上,也不需要用户定义数据挖掘模式,同时提出了将三元组中的"object+predict"作为关联规则中的"项",在属于同一 subject 的三元组中发现关联规则,这使得所生成的关联规则更容易理解。还有一些方法直接作用在 RDF 三元组上[6,7],一个三元组就是一条事务记录,分别将 subject,predict,object 作为事务 ID,从而可以产生 6 种发现关联规则的配置,但这类方法忽略了 RDF 数据集中蕴含的领域本体中描述的背景知识的作用,也没有关注算法的执行效率。

由于 RDF 数据也可以通过 RDF 图的方式呈现,因此有研究将关联数据集转化为边带有标记的有向图,通过图挖掘方法从中找到频繁子图或者频繁子树,从而发现频繁出现的 RDF 数据[8,9]。由于图挖掘主要作用于 RDF 图的拓扑结构,关联数据集的语义信息也被忽略,因此难以发现语义上相关的内容,对于所发现的频繁子图或子树结构,也难以将其转化为用户容易理解的关联规则的形式。目前也出现了可用于关联数据挖掘的较为完整的数据挖掘系统,如 LiDDM 和 RapidMiner 等[10,11]。LiDDM 可实现关联数据集上的关联规则挖掘、分类、聚类等任务,用户首先查询 LOD 数据集获取待挖掘的数据,LiDDM 将查询结果转化为传统数据挖掘算法可处理的格式;RapidMiner semweb plugin 与 LiDDM 类似,接收 SPARQL 查询结果作为输入,再将其转化为经典算法可处理的格式。这类工具除了处理效率难以满足关联数据的要求外,也难以处理数据集之间领域本体的差异和不一致。SPARQL-ML 作为 SPARQL 查询语言的扩展,对于特定的语义 Web 数据可以生成针对指定概念的数据挖掘模型,但它也仅适用于单个数据集且功能十分有限,目前只能完成分类和预测任务[12]。

考虑到关联数据的大数据特性,一些研究使用 Hadoop 技术实现语义大数据上的查询与推理[13,14],Hadoop 可以满足关联数据的分布式特性并可明显提高数据处理的效率;大量的研究也成功地将经典的关联规则挖掘算法部署在 Hadoop 环境下。这些研究为解决关联数据挖掘的效率问题提供了有价值的借鉴,但仍然没有涉及到如何选择合适的数据集合、如何处理关联数据集合间的异构和冲突等问题。

通过以上分析可以发现,现有的相关研究还没能很好地解决多个相关联、异构的 RDF 数据集上的有效的关联规则挖掘的问题,其主要问题在于忽视了关联数据集之间的链接关系和领域本体对关联规则挖掘的影响。本文研究将针对目前研究成果在这方面的不足,提出具体的解决方案。

3 关联数据的模式级链接

本文将通过建立关联数据的模式级链接的方式实现 LOD 上的关联规则的挖掘,因为建立模式级链接将很大程度上降低同领域内多个异构 RDF 数据集对数据挖掘所产生的影响。为了对关联数据的模式级链接进行准确的描述并方便下文的论述,首先对文中出现的主要概念进行形式化的说明。

RDF 三元组(RDF Triple):给定一个 URI 集合 R、空结点集合 B、文字描述集合 L,一个 RDF 三元组 t 是形如(s,p,o)的三元组,其中 $s\in R\cup B$,$p\in R$。这里的 s 通常称为主语(subject)、资源(resource)或主体,p 称为谓词(predicate)或属性(property),o 称为宾语(object)、属性值(value)或客体。后续小节使用 s,p,o 分别表示组成三元组的主语、谓词和宾语。

领域本体(Domain Ontology):一个本体 O 可表示为$\langle C,P,R,I,A\rangle$,其中 $C=\{c_1,c_2,\cdots,c_n\}$ 为概念集合;$P=\bigcup_{i=1}^{n}P_i$,$P_i=\{p_1,p_2,\cdots,p_m\}$ 为某个概念 c_i 的属性集合;$R=\bigcup_{i=1}^{n}R_i$,$R_i=\{r_1,r_2,\cdots,r_k\}$ 为某个属性 p_j 上的约束集合,对 p_j 的取值进行限制;$I=\bigcup_{i=1}^{n}I_i$,I_i 为概念 c_i 的实例集合;$A=\bigcup_{i=1}^{n}A_i$,$A_i=\{a_1,a_2,\cdots,a_l\}$ 为公理的集合,说明了某个属性的性质。本研究假设本体以 OWL 语法形式进行描述。

本体 O 也可简化表示为$\langle S,A\rangle$,其中 $S=C\cup P\cup R\cup I$ 为本体中出现的所有词汇的集合,A 为公理集合。

RDF 数据集(RDF Data Sets):设 $OS=\{O_1,O_2,\cdots,O_m\}$ 为领域本体集合,一个特定领域上的 RDF 数据集 RS 是一个 RDF 三元组集合,对于集合 RS 中的每个三元组(s,p,o),$\exists O_i\in OS$ 使得 s 为 $O_i(C)$ 中某个概念的实例或为空结点,p 为 $O_i(P)$ 中的某个属性。

关联数据集(Linked Open Data Sets):关联数据集 $LD=\{RS_1,RS_2,\cdots,RS_k\}$ 是一组特定领域上的 RDF 数据集 RS 的集合,$\forall RS_i\in LD$ 都可以通过 HTTP 协议访问;并且存在谓词 p,使得通过 p 可以实现从 RS_i 到 RS_j 的直接访问。

可以看到,关联数据集包含多个 RDF 数据集,涉及多个领域本体,在 RDF 数据集之间建立模式级链接就是在各自所遵循的领域本体之间建立对应关系,理论上本体映射技术可用于实现模式级链接的建立:对于数据挖掘任务所考虑的多个 RDF 数据集合,建立多个领域本体中概念、属性及实例间的等价关系和层次关系。关联数据模式级链接需要在多个本体之间建立相互的映射关系,这个过程等价于建立多次两个本体之间的映射。据此关联数据的模式级链接可定义如下。

定义 1(关联数据的模式级链接) 设 $OS=\{O_1,O_2,\cdots,O_m\}$ 为领域本体集合,关联数据集 $LD=\{RS_1,RS_2,\cdots,RS_n\}$ 是以 $O_i(O_i\in OS)$ 为参照本体的 RDF 数据集集合,则 LD 中数据集的模式级链接可表示为函数型映射集合 $F=\{f_1,f_2,\cdots,f_k\}$,$f_i:S_s\rightarrow S_t$,$S_s\in O_s$,$S_t\in O_t$,S_s,S_t 分别为本体 O_s,O_t 中的词汇集合。其中映射 $f_i:S_s\rightarrow S_t$ 可作用在概念、属性、实例上。

基于本体映射技术建立关联数据模式级链接的具体方法不是本文研究的重点内容,在此不进行分析。

4 关联数据集上的关联规则挖掘

众所周知，频繁项集的产生是关联规则挖掘的关键步骤，本文以关联数据集上频繁项集的生成方法作为研究重点，首先提出了单个 RDF 数据集上的频繁 RDF 项集生成方法，再将其扩展到建立了模式级链接的多个 RDF 数据集上。由于 RDF 数据模式与经典数据挖掘方法所处理的数据格式差异巨大，首先对 RDF 数据集上与频繁项集相关的概念进行定义，定义中使用了描述逻辑(Description Logic，DL)中的关系描述符号和构词符号，关于描述逻辑的语法规则不在此赘述。

4.1 与 RDF 项集相关的基本定义

定义 2(属性链) 本体 O 上的一个属性链 $pl=(p_1\circ p_2\circ\cdots\circ p_n)$是一条从概念 c_i 到概念 c_j 的一条路径当且仅当 $O\vdash c_i\sqsubseteq\exists p_1\circ p_2\circ\cdots\circ p_n\cdot c_j$，从概念 c_i 到概念 c_j 的路径集合记为 $Path(c_i,c_j)$。其中 $c_i,c_j\in O(C),p_1,p_2,\cdots,p_n\in O(P)$。

属性链也可作用于概念实例。$Instance(c)$表示概念 c 的实例集合，设 $ins_i\in Instance(c_i),ins_j\in Instance(c_j)$，则 $Path(ins_i,ins_j)$表示从 ins_i 到 ins_j 所经历的路径。

定义 3(RDF 挖掘模式) 给定领域本体 O，RDF 数据集 RS 上的数据挖掘模式 Q 是一个二元组 $(C_{key},C_{related})$，其中 C_{key} 是用户关注的核心概念，$C_{related}$ 是通过属性链可以和 C_{key} 建立关联的所有的概念集合，$C_{key}\in O(C),C_{related}\in O(C)$。$C_{key}$ 和 $C_{related}$ 都是在本体 O 中定义的概念，其间具有清晰的层次关系，具体可表现为：$\forall c\in C_{related}$ 有 $c\sqsubseteq C_{key}$。

定义 4(RDF 数据项模式与 RDF 数据项) 给定一个数据挖掘模式 Q，RDF 数据项模式 q 具有如下形式：$q=c_r(X)\wedge\alpha_1\wedge\cdots\wedge\alpha_n,c_r\in C_{related}$，其中 X 是模式中唯一受 c_r 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。将 RDF 数据项模式 q 中的变量替换为概念实例就构成了数据项模式的一个实例，称为 RDF 数据项。

定义 5(RDF 数据事务) 给定 RDF 数据挖掘模式 Q，ITE 为 Q 上的数据项的集合，Q 上的一条关于概念 $c_i(c_i\in C_{related})$的实例 $instance_i$ 的事务 T 是满足如下条件的 RDF 数据项集合：

(1) $\forall t\in T,\exists c'\in C_{related},O\vdash c_i(t)\wedge c_i\sqsubseteq c'$；

(2) $\exists ins_k\in instance(c_j)$，满足 $\exists p\in path(t,ins_k)\wedge\exists p'\in path(ins_k,key)$，其中 $c_j\in C_{related},key\in Instance(C_{key})$。

据此定义，在 RDF 数据集中，事务 ID 使用概念实例 $instance$(对应于 RDF 三元组中的主语或者宾语)进行标示，一条事务是与 $instance$ 相关的所有 RDF 数据项。

在以上定义的基础上，可以对 RDF 数据集上的关联规则挖掘进行描述：给定领域本体 O、RDF 数据挖掘模式和 RDF 事务集合，发现满足支持度阈值 s 和置信度阈值 c 的关联规则。符合定义 5 的事务集合与传统的事务数据库具有相似的结构，理论上可使用经典的关联规则挖掘方法处理 RDF 事务数据集合。

例 1 图 1 对应的 RDF 数据集上的事务举例。

分析概念“*Student*”、“*Teacher*”的实例及其上的一组属性。其中概念“*Student*”的实例有“*Zhang*”、“*Wang*”和“*Yuan*”，概念“*Teacher*”的实例有“*Prof. Li*”、“*Prof. Wu*”和“*Prof. Hu*”；实例具有的属性以及实例间的关系如图 1 所示。

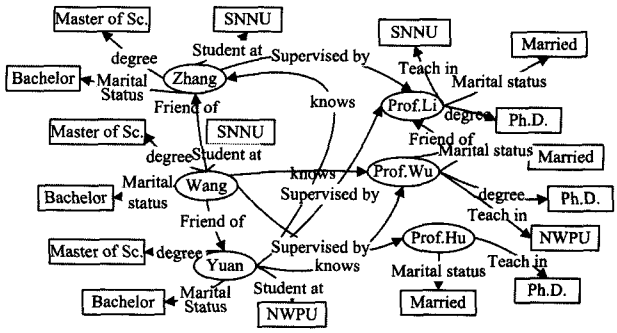

图 1 包含概念“Student”、“Teacher”及其属性的 RDF 数据图

假设用户关注的核心概念为“*Student*”，$Student(X)$，$Student(X)\wedge SupervisedBy(X,Y)$，$Student(X)\wedge Knows(Z,X)$，$Student(X)\wedge StudyAt(X,S)$，$Student(X)\wedge Degree(X,W)$，$Student(X)\wedge Knows(Z,X)\wedge Degree(X,W)$都属于 RDF 数据项模式；$Student(Zhang)$，$Student(Zhang)\wedge SupervisedBy(Zhang,Prof.\ Li)$，$Student(Zhang)\wedge Knows(Yuan,Zhang)$，$Student(Zhang)\wedge StudyAt(Zhang,SNNU)$，$Student(Zhang)\wedge Degree(Zhang,"Master\ of\ Sc.")$，$Student(Zhang)\wedge Knows(Yuan,Zhang)\wedge Degree(Zhang,"Master\ of\ Sc.")$都属于 RDF 数据项。根据定义 4，基于该 RDF 图的事务集合具有表 1 的形式。

表 1 图 1 对应的 RDF 数据事务举例

Transaction ID	Transaction
Zhang	StudentAt(SNNU), Knows(Yuan), Supervisedby(Prof. Li), Knows(Prof. Li) ∧ Degree("Master of Sc.") ∧ StudentAt(SNNU),…
Wang	StudentAt(SNNU), Knows(Prof. Wu), Supervisedby(Prof. Hu), Knows(Prof. Wu) ∧ Degree("Master of Sc.") ∧ StudentAt(SNNU), Friendof(Wang) ∧ MaritalStatus(Bachelor),…
…	…

4.2 单个 RDF 数据集上 RDF 数据项的生成

RDF 数据项是实例级的数据，只要能够发现数据项实例所遵循的模式，生成其实例数据就容易实现。对此，本文提出基于数据项模式的 RDF 数据项生成方法。

4.2.1 与 RDF 数据项模式相关的概念

定义 6(RDF 数据项模式间的关系) 设 q,q'为数据挖掘模式 Q 上关于概念 C_{key} 的任意两个数据项模式，$answerset(q,RS)$表示在 RS 中满足模式 q 的数据项集合，若 $answerset(q,RS)\subseteq answerset(q',RS)$，则称 q 为比 q'更一般的模式，表示为 $q\preceq q'$。

定义 7(RDF 数据项模式间的关系) 设 q,q'为数据挖掘模式 Q 上关于概念 C_{key} 的任意两个数据项模式，称 q 是 q'的祖先(或后代)，当 q 为将 q'中出现的某个谓词 p'用谓词 p 来替换得到的模式，p 满足 $p'\sqsubseteq p$(或 $p\sqsubseteq p'$)。若 q 为 q'的祖先，则 $q\preceq q'$。

定义 8(RDF 数据项模式支持度) 设 $q=c_r(X)\wedge\alpha_1\wedge\cdots\wedge\alpha_n$为数据挖掘模式 Q 上关于概念 C_r 的数据项模式，q 在数据集 RS 中的支持度 $supp(q,RS)$定义为：

$$supp(q,RS)=\frac{|answerset(q,RS)|}{|answerset(\hat{q},RS)|},\hat{q}=c_r(X)$$

定义 9(RDF 数据项频繁模式) 称一个具有支持度 s 的数据挖掘模式 Q 中的 RDF 数据项模式 q 在 RS 中是频繁的，如果满足 $s\geqslant minsup$，其中 $minsup$ 为支持度阈值。

显然，如果 q 是频繁的，q 的所有祖先也是频繁的。

定义 10（k-模式） 称模式 $q=c_r(X)\wedge\alpha_1\wedge\cdots\wedge\alpha_n$ 为 k-模式，如果其中除 $c_r(X)$ 外，还包含 k 个不同谓词。

4.2.2 RDF 数据项模式生成规则

事实上，定义 4 数据项模式 q 中的变量 X 对应着 RDF 数据集上的概念实例，变量 X 取相同值的一组模式实例集合就构成了一条事务。因此，只要找到关于挖掘模式 Q 的所有的数据项模式，生成数据项的任务就可通过查询 RDF 数据集的方式实现，各种查询优化策略[15,16]可用于数据项的生成过程以提高效率。本研究提出了以下规则用于数据项模式的产生。将以模式 $q=C_{key}(X)\wedge c(Y)\wedge p(X,Y)$（$C_{key}$ 为用户关注的概念，$c\in C(O)$，$p\in P(O)$）为例，对这些规则进行说明。

RF(1)（概念层次规则） 将 q 中的概念用明确断言的其子概念替换；对 q 使用 RF1，可得到模式 $q_1=C_{key}(X)\wedge c_{sub}(Y)\wedge p(X,Y)$，其中 c_{sub} 为 c 的子概念。

RF(2)（属性层次规则） 将 q 中的属性用明确断言的其子属性替换；对 q 使用规则 RF2，则生成模式 $q_2=C_{key}(X)\wedge c_{sub}(Y)\wedge p_{sub}(X,Y)$，其中 p_{sub} 为 p 的子属性。

RF(3)（谓词添加规则） 向模式 q 中添加新的概念或属性，最多向模式中引入一个新的约束变量；对 q 使用 RF3，添加属性 p_1，可得到模式 $q_3=C_{key}(X)\wedge c(Y)\wedge p(X,Y)\wedge p_1(X,Z)$，其中对变量 X 增加了新的约束，并引入一个新变量 Z。

RF(4)（谓词拷贝规则） 对于已经出现在 q 中的某个概念或属性，向 q 添加其拷贝，约束其它变量，同时引入最多一个新的约束变量。对 q_3 利用 RF4 可得到 $q_4=C_{key}(X)\wedge c(Y)\wedge p(X,Y)\wedge P(X,Z)\wedge p_1(X,Z)$。

其中，RF(1)，RF(2)是将模式 q 用其后代来替换，在 k 模式 q 上使用这两个规则，产生的仍为 k 模式；RF(3)向模式中加入新的概念或属性；RF(4)向模式中加入谓词的拷贝，在 k 模式 p 上使用 RF(3)，RF(4)产生 $k+1$ 模式，并且可能带来新的约束变量，每次最多引入一个新的约束变量，模式中允许包含常量。

数据项模式生成规则具有如下性质，可用于提高数据项模式生成的效率。该性质保证仅对频繁模式施加求精规则才可能产生频繁模式，对非频繁模式不必对其使用求精规则。

性质 1 设 q 为数据挖掘模式 Q 上关于概念 C_{key} 的任意两个数据项模式，若一个 k-模式 q 是频繁的，则以下两者须满足其一：

(1)任一个包含 q 的 $k-1$ 模式是频繁的；

(2)q 的祖先 k-模式 q' 是频繁的。

该结论的证明是容易的。情形(1)类似于 Aprior 定理所表达的“任何非频繁 $k-1$ 项集都不可能是频繁 k 项集的子集”的论断。一个 $k-1$ 模式 q' 通过使用求精规则 RF(3)或 RF(4)演化为 k 模式 q，q' 较 p 更一般，若 q' 非频繁，则 q 也必定非频繁。由于 q 的祖先是较 q 更一般的模式，情形(2)可根据频繁模式的性质得到。

若一个 k-模式 q 是非频繁的，则使用求精规则 RF(3)，RF(4)产生的 $k+1$ 模式 q' 也是非频繁的；使用求精规则 RF(1)，RF(2)产生的 k 模式 q' 也是非频繁的。一般地，每产生一个候选模式，都需要测试其是否已经被一个已知的非频繁模式所包含。如果是，则该模式必定为非频繁模式，不需要进一步对其进行评价，也不需要对其使用任何求精规则。

4.2.3 单个 RDF 数据集上关联规则的产生

RDF 数据项可看作对多个谓词使用合取运算符构成的子句。RDF 关联规则描述的是 RDF 数据项各个谓词之间的关系，并且这些谓词所约束的是具体的概念实例以及实例上的属性取值。尽管 RDF 数据项频繁模式一定程度上也表达了谓词之间的关系，但这种关系是模式级的，对于用户而言不够直观、不易理解，更有价值的关联规则是实例级的谓词之间的关系。

定义 11（频繁 RDF 数据项集） 设 RDF 数据项频繁模式 q 为 k 模式，$ITEM_q$ 为满足 q 的 RDF 数据项的集合，则称 $ITEM_q$（$ITEM_q=SI_1\cup SI_2\cup\cdots\cup SI_r, SI_i\cap SI_j=\varnothing$）为 RDF 数据集上的候选频繁 k-RDF 数据项集，对于 $\forall item_1, item_2\in SI_i$，$item_1, item_2$ 满足除了用户关注概念 C_{key} 所约束的实例不同外，$item_1, item_2$ 中的其它部分完全相同。若 $\frac{|SI_i|}{|ITEM_q|}\geqslant f$，$f$ 为支持度阈值，则称 SI_i 为频繁 RDF 数据项集。

例 2 图 1 对应的 RDF 数据集上的关联规则举例。

从图 1 中可得到 RDF 数据项模式 $q=Student(X)\wedge Knows(X,Y)\wedge Degree(X,Z)\wedge StudentAt(X,S)$，该模式为 3 模式，在表 1 中该模式对应的 RDF 数据项有 2 个，可划分为各包含一个数据项的两个集合{$Knows(Prof.\ Li)\wedge Degree(Master\ of\ Sc.)\wedge StudentAt(SNNU)$}和{$Knows(Prof.\ Wu)\wedge Degree("Master\ of\ Sc.)\wedge StudentAt(SNNU)$}。显然两个数据项的支持度均为 0.5。

从频繁 RDF 数据项集可以直接构造关联规则。例如，对 RDF 数据项 $Knows(Prof.\ Li)\wedge Degree("Master\ of\ Sc.)\wedge StudentAt(SNNU)$ 可构造如下规则：

$StudentAt(SNNU), Knows(Prof.\ Li)\rightarrow Degree(Master\ Sc.)$

$Knows(Prof.\ Li), Degree(Master\ Sc.)\rightarrow StudentAt(SNNU)$

显然，可以从 RDF 数据项频繁模式 q 产生的 RDF 数据项中直接构造关联规则，在这一过程中并没有明显地将 RDF 数据集改造成如定义 5 所描述的 RDF 事务格式，但是在定义 11 中，对候选频繁项集进行分割的过程遵循了定义 5 的事务格式，其目的是为了将 RDF 数据项以一种清晰、容易理解的方式呈现。这种呈现方式充分利用了 RDF 数据集中蕴含的语义信息，同时经典的关联规则挖掘算法也可作用于定义 5 的格式以协助发现关联规则。

4.3 模式级链接关联数据集上频繁 RDF 项集的发现

4.3.1 扩展 RDF 数据项模式生成规则

在建立了关联数据的模式级链接之后，就可以定义多个 RDF 数据集合上的频繁模式发现过程。本节将扩展 4.2 节提出的单个 RDF 数据集上的频繁模式发现方法，使其可以作用于多数据集。显然，当同时考虑多数据集时，就需要对数据项模式的生成规则进行相应的扩充。

设已建立模式级链接的关联数据集所涉及到的领域本体集合为 $OS=\{O_1, O_2, \cdots, O_m\}$，对于 $\forall c_i\in C(O_i)$，$Eq(c_i)$ 表示与 c_i 等价的 OS 中的概念集合，$Sub(c_i)$ 表示 c_i 的子概念集合。类似地，对于 $\forall p_i\in P(O_i)$，$Eq(p_i)$ 表示与 p_i 等价的 OS 中的属性集合，$Sub(p_i)$ 表示 p_i 的子概念集合。仍然以基于 O_i 的模式 $q=C_{key}(X)\wedge c_i(Y)\wedge p_i(X,Y)$（$C_{key}$ 为用户关注的概念，$c_i\in C(O_i)$，$p_i\in P(O_i)$）为例，对扩展的数据项模式生成

规则进行说明。具体规则如下：

ERF(1)(概念等价规则)：将 q 中的概念用 $Eq(c_i)$ 中的概念替换；对 q 使用 ERF(1)，可得到模式 $q_1=C_{key}(X)\wedge c_{eq}^k(Y)\wedge p_i(X,Y)$，其中 c_{eq} 为第 k 个数据集中 c_i 的等价概念。

ERF(2)(概念层次规则) 将 q 中的概念用 $Sub(c_i)$ 中的概念替换；对 q 使用 ERF(2)，可得到模式 $q_2=C_{key}(X)\wedge c_{sub}^k(Y)\wedge p_i(X,Y)$，其中 c_{sub} 为第 k 个数据集中 c_i 的子概念。

ERF(3)(属性等价规则)：将 q 中的属性用 $Eq(p_i)$ 中的子属性替换；对 q 使用规则 ERF(3)，则生成模式 $q_3=C_{key}(X)\wedge c_i(Y)\wedge p_{eq}^k(X,Y)$，其中 p_{eq} 为第 k 个数据集中 p_i 的等价属性。

ERF(4)(属性层次规则) 将 q 中的属性用 $Sub(p_i)$ 中的子属性替换；对 q 使用规则 ERF(4)，则生成模式 $q_4=C_{key}(X)\wedge c_i(Y)\wedge p_{sub}^k(X,Y)$，其中 p_{sub} 为第 k 个数据集中 p_i 的子属性。

ERF(5)(谓词添加规则) 向模式 q 中添加一个新的 OS 中的概念或属性，最多向模式中引入一个新的约束变量；对 q 使用 ERF(5)，添加属性 p'，可得到模式 $q_5=C_{key}(X)\wedge c_i(Y)\wedge p_i(X,Y)\wedge p_j^k(X,Z)$，使用 p_j^k，第 k 个数据集中的属性对变量 X 增加了新的约束，并引入一个新变量 Z。

ERF(6)(谓词拷贝规则) 对于已经出现在 q 中的某个概念或属性，向 q 添加其拷贝，约束其它变量，同时引入最多一个新的约束变量。对 q_5 利用 ERF(6)，p_i 增加对变量 Z 的约束，可得到模式 $q_6=C_{key}(X)\wedge c_i(Y)\wedge p_i(X,Y)\wedge p_i(X,Z)\wedge p_j^k(X,Z)$。

与 4.2 节的规则相比，扩展的 RDF 数据项模式生成规则 ERF(1)、ERF(3)都可能生成与原模式在语义上等价的新模式；其次这些规则能在本体映射结果的基础上找到关联数据集上所有可能的数据项模式。对基于 O_i 的模式 q 而言，由于 ERF(5)的效用，从 q 所派生出的模式可能包含属于不同领域本体的概念或属性，这也是关联数据上的数据挖掘可以从单个数据集扩展到同领域数据集的主要原因。

例如，对于图 1 表示的 RDF 数据集合，存在另一同样以“*Student*”为主要概念的 RDF 数据集，在该集合中“*Student*”上定义有图 1 中不存在的属性“*HealthStatus*”，这样，模式 $q=Student(X)\wedge Knows(X,Y)\wedge Degree(X,Z)\wedge StudentAt(X,S)$ 就有可能被扩展为 $q'=Student(X)\wedge Knows(X,Y)\wedge Degree(X,Z)\wedge StudentAt(X,S)\wedge HealthStatus(X,H)$。

为了获得模式 q 对应的数据项，对于从 q 扩展出的数据项模式集合 MQ，将其划分为 n 个集合，即 $MQ=MQ_1\cup MQ_2\cup\cdots\cup MQ_n$，这样每个数据集都有相应的 MQ_i，根据该数据项模式集合从对应数据集中查找 RDF 数据项。算法 1 是关联数据集上生成 RDF 数据项模式的伪代码描述。

算法 1 关联数据集上数据项模式生成

输入：数据挖掘模式 Q，关联数据集 LD={RS_1，RS_2，…，RS_n}，领域本体集合 OS={O_1，O_2，…，O_m}上的映射结果 F

输出：RDF 数据项模式集合 MQ_i，i=1，…，n

```
1. 根据 Q,定义初始 q=C_key(X)
2. MQ_i=∅,i=1,…,n;//每个数据集的数据项模式集合初始化为空集
3. MQ={q};//使用规则所新产生的数据项模式放在集合 MQ 中
4.   for each (q∈MQ and not ApplyItemSchemaGenRules(q) and
     Frequent(q))//对集合 MQ 中所有没有施加规则的数据项频繁
     模式应用数据项模式生成规则
5.     for (i=1;i<=6; i++)
6.       if New(ApplyERF(i)(q)) MQ=MQ∪{ApplyERF(i)
         (q)}//如果使用规则 ERF(i)作用于 q 可以产生新的有意
         义的模式,将 q 放入 MQ
7.   Divide(MQ);//将 MQ 根据模式中概念和属性所属的数据集进
     行分割
8.   RETURN (MQ_i,i=1,…,n)
```

算法 1 不需要处理海量的实例数据，该过程与建立多个本体之间的映射关系一样，可作为数据预处理阶段的成果。

4.3.2 关联数据集上的频繁 RDF 数据项发现及关联规则的生成

在分别获得了每个数据集上数据项模式集合 MQ_i 之后，多个数据集上的频繁 RDF 数据项发现与单个 RDF 数据集具有相似的步骤，主要的差别表现在两点：1)数据项模式支持度的计算；2)以概念实例为 ID 的事务生成。关联数据集上，数据项模式支持度的计算遵循如下定义。

定义 12(关联数据集上 RDF 数据项模式支持度) 关联数据集 LD 中，模式 q 的支持度 $supp(q,LD)$ 定义如下：

$$supp(q,LD)=\frac{\left|\sum_{q_i\in EQ(q)}\sum_{RS_i\in LD}answerset(q_i,RS_i)\right|}{\left|\sum_{RS_i\in LD}answerset(\hat{q},RS_i)\right|},$$

$$\hat{q}=c_r(X)$$

其中，$EQ(q)$ 表示与 q 等价的模式的集合。

关联数据集上 $supp(q,LD)$ 的计算将同时考虑与 q 语义等价的所有模式在关联数据每个数据集上的实例数量。

对于以概念实例为 ID 的事务，也会考虑概念的等价关系。等价的概念所对应的实例也是等价的，对此，本文采取的策略是将语义等价的概念实例对应的事务进行合并。具体方法是将等价的概念实例标记的数据项集合进行合并，并选择其中一个概念实例作为事务 ID。这种策略的结果也会产生一条事务记录中包含来自不同数据集的 RDF 数据项的现象，这些 RDF 数据项属于共同的领域，因此关联数据集上的关联规则挖掘从面向单一数据集扩展到同一领域内的数据集，且关联规则的生成方式与 4.2.3 节中提到的方式完全相同。

5 基于 Hadoop 的模式级链接关联数据的频繁模式挖掘的实现

为了处理海量的关联数据，本研究使用 Hadoop 相关技术验证上文提出的思想。可以看到，概念实例的查询和数据项模式支持度的计算是频繁模式发现过程中的关键任务，这些关键任务都使用 MapReduce 计算模型来实现。

5.1 概念实例的查询

查询概念实例是实现本体映射和数据项模式频繁度计算的基础，算法 2 是实现该过程的 MapReduce 伪代码实现。该算法将返回一个概念的所有实例。

算法 2 查询概念实例的 MapReduce 算法伪代码

```
mapper(key,value)
//key:concept
//value:triple
begin
  if (triple.predicate==rdf.type && triple.object==concept)
    output(concept,triple.subject);
end
reducer (key,iterator values)
begin
List (triple) list=new List (triple) ();bbbv
```

```
for (rdfnode in values)
  list. add(rdfnode );
output(concept,rdfnode);
end
```

5.2 数据项模式 q 支持度的计算

由于数据项模式 q 也可以理解为作用在关联数据集上的连接查询，因此可以将连接查询 q 分解为若干子查询 q_1，q_2，…，q_m，满足 $q_1 \wedge q_2 \wedge \cdots \wedge q_m = q$。假设待查询的 RDF 数据集合 RS 被分割为 n 个子集，即 RS_1，RS_2，…，RS_n，设 $t(RS_i)$ 表示 RS_i 中的三元组，$nd(RS_i)$ 表示 RS_i 中的节点，则有 $t(RS_i) \cap t(RS_j) = \emptyset$，定义 $B(RS_i)$ 为 RS_i 的边界节点，则对于 $\forall n \in B(RS_i)$ 有 $\exists RS_j\ (i \neq j)$ 使得 $n = nd(RS_i) \cap nd(RS_j)$。则从 RS 中查询满足 q 的查询变量的过程可以分为两个阶段：首先在 LD 中找到分别满足子查询 q_1，q_2，…，q_m 的变量集合 V_1，V_2，…，V_m；其次对 V_1，V_2，…，V_m 进行合并，最终获得满足查询 q 的变量集合 V。

设关联数据集上的数据项模式集合为 $QM = \{q_1, q_2, \cdots, q_m\}$，则对 $\forall q_i \in QM$ 都可获得满足 q_i 的一组变量集合 V_i。对于形如 $q_i = C_{key}(X) \wedge c(Y) \wedge p(X,Y)$ 的数据项模式，有 $V_i = \{(x_1, y_1), (x_2, y_2), \cdots, (x_l, y_l)\}$，$|V_i| = l$ 表示满足 q_i 的变量组数量。根据 V_i 可得到 q_i 对应的 RDF 数据项集合 $QI_i = \{C_{key}(x_1) \wedge c(y_1) \wedge p(x_1, y_1), C_{key}(x_2) \wedge c(y_2) \wedge p(x_2, y_2), \cdots, C_{key}(x_l) \wedge c(y_l) \wedge p(x_l, y_l)\}$。若 $|QI_i| \geqslant f$，f 为用户定义的频繁度阈值，则 q_i 就认为是数据项频繁模式，根据算法 1 就能够对 q_i 施加规则进而产生新模式。

若 RDF 数据项模式包含了属于不同数据集的概念或属性，如 $q = C_{key}(X) \wedge c^1(Y) \wedge p^2(X,Y)$，其中 c，p 分别属于不同的数据集，则可将 q 分解为两个模式 $q_1 = C_{key}(X) \wedge c^1(Y)$，$q_2 = C_{key}(X) \wedge p^2(X,Y)$，分别查询这两个模式的数据项，再对结果以 X 为中介进行合并。

在这个过程中，频繁数据项模式的筛选理论上需要查询整个 RDF 数据集，但是根据性质 1，$k+1$ 模式的频繁度计算可以建立在 k-模式的基础上，也就是说只需要查询满足 k-模式的数据集合，而不需要查询整个 RDF 数据集。相比直接在整个数据集上查找 RDF 频繁数据项，性质 1 的使用将在很大程度上减少作用在整个数据集上的计算任务。

5.3 RDF 数据项事务集合的生成

为了充分利用已经相当成熟的频繁模式发现和关联规则挖掘成果，需要把 $QI = \{QI_1, QI_2, \cdots, QI_m\}$ 转变为传统的事务数据库格式。根据定义 5，RDF 数据项事务 ID 使用概念实例 *instance* 进行标示，一条事务是与 *instance* 相关的所有 RDF 数据项。使用算法 2 得到用户感兴趣的核心概念 C_{key} 的实例集合 $I(C_{key})$ 之后，对于 $\forall i \in I(C_{key})$，QI 需要找到与其相关的所有数据项 $item_i \in QI$，从而产生一条事务。该过程也可利用 MapReduce 框架实现，算法 3 是其伪代码实现。

算法 3 生成与实例 *instance* 相关的 RDF 数据项

```
mapper (key,value)
//key:(i,q)
//value:RDFitem
begin
  if (i is an instance of a concept in q||i is an instance of Domain(P)
  or Range(P))
    output(i,RDFitem);
end
reducer (key,iterator values)
//key:i
begin
List (RDFitem) list=new List (RDFitem) ();
for (RDFitem in values)
  if (i appeared in RDFitem )
    output(i,RDFitem);
end
```

6 实验

本文实验的主要目的在于验证模式级链接的建立对关联数据集上数据挖掘结果的影响，以及基于 RDF 数据模型语义特征的频繁项集生成过程对关联规则的影响，其次对文中涉及算法的有效性进行验证。实验基于 Hadoop 技术，以 HDFS 形式存储各个关联数据集、RDF 数据项模式集合、RDF 数据项集合以及 RDF 数据项事务集合，在此基础上使用 MapReduce 计算模型处理 RDF 上的查询和与 RDF 数据项有关的计算。实验的硬件环境为一个 14 节点的 Hadoop 集群，每个节点配置均为 Intel Pentium(R) Dual-Core CPU E5700 3.0GHz,4GB RAM。

(1)模式级链接对 RDF 数据项模式的影响

本文研究所采用的数据集均来自于 LOD，如表 2 所列。

表 2 关联数据集以及对应的模式

模式名称	采用该模式的 LOD 数据集	模式中定义的“类”的个数	通过实例链接的 LOD 数据集举例
DBpedia[1)]	DBpedia	685	Geonames, US Census
Geonames[2)]	Geonames, Geospecies	11	DBpedia, Jamendo, FOAF Profiles
Music Ontology[3)]	Jamendo, Music Brainz, DBTunes	136	GovTrack, DBpedia, Geonames
BBC Program[4)]	BBC Programs BBC Music	100	BBC Music, BBC Playcount Data
FOAF Profiles[5)]	FOAF, Music Brainz	16	Crunch Base, SIOC Site,
SIOC[6)]	DBpedia, LinkedMDB	14	SmanticWeb. org, FOAF Profiles
AKT Reference Ontology[7)]	ACM DBLP	17	Pisa, IEEE, eprints
Semantic Web Conference Ontology[8)]	SW Conference Corpus	177	SmanticWeb. org, Revyu

1) http://wiki.dbpedia.org/Downloads351/#dbpediaontology

2) http://www.geonames.org/ontology

3) http://musicontology.com

4) http://purl.org/ontology/po

5) http://xmlns.com/foaf/spec

6) http://rdfs.org/sioc/ns#

7) http://www.aktors.org/ontology/support

8) http://data.semanticweb.org/ns/swc/ontology

这些数据集来自多个不同的领域，其所遵循的模式都具有一定的代表性。为了验证同领域数据集的模式级链接对频繁模式发现的影响，在同领域内又选择如下几组本体建立模式级链接。D1：Music Ontology 与 BBC Program；D2：Music Ontology 与 DBpedia；D3：FOAF Ontology 与 DBpedia；D4：Geonames 与 DBpedia；D5：FOAF 与 SIOC；D6：The Semantic-Web Conference Ontology 与 the AKT Reference Ontolog。同时在相应的数据集上实现模式级链接的关联规则挖掘。由于本体映射的准确程度直接影响到模式级链接的结果，为了准确地体现关联数据模式级链接数据挖掘与单个数据集上数据挖掘结果的差异，本研究使用人工方式实现本体映射。对于表 2 中所涉及的本体概念、属性，根据相应规范中对概念和属性的解释说明，人工建立映射关系以保证模式级链接结果的准确性。

在不限定频繁度的情况下，本研究从 D1－D6 每个领域中分别选择了两个大小约为 600000 条的 RDF 三元组集合，从所产生 RDF 事务的数量和事务平均包含的 RDF 数据项个数上对单个数据集和模式级链接的数据集上的频繁模式挖掘结果进行比较，结果如表 3 所列。其中领域：数据集表示在特定领域内所选择的两个不同的数据集，数据集名称之间用“/”隔开，括号中的概念表示频繁模式挖掘过程中对应数据集中用户关注的概念，如果用户关注的概念在其中某个数据集中不存在，则利用本体映射的方法从中找到对应的概念，用户关注概念名称用斜体表示。所生成的 RDF 数据项模式数量分为两类，分别在单个数据集上生成的模式和建立模式级连接后每个数据集上模式的数量。可以看出，在没有建立模式级链接的情况下，如果在数据集中不能直接找到用户关注概念，则无法产生用户可能感兴趣的数据项模式，模式级链接的建立能够通过扩展等价概念或属性有效地帮助用户找到所关注的模式。例如数据集 D1：Music Brainz/BBC Music，BBC Music 遵循的本体是 BBC Program，其中并没有 MusicArtist 这个用户关注概念，建立模式级链接的过程发现 BBC Music 在描述数据时使用了 Music Ontology，因此具有与 Music Brainz 相同的数据项模式集合；对于 D3：FOAF Profiles/DBpedia (Person/Person)和 D4：Geonames/DBpedia，尽管用户关注概念在 DBpedia 中存在，但 DBpedia 中相应概念的属性数量较少，模式级链接可以对概念具有的属性进行扩展，从而能够更深层次地发现满足用户要求的模式。

表 3　不同领域模式级链接与非模式级链接数据集所产生的事务数量

领域：数据集	RDF 数据项模式数量		所产生的事务数量	事务平均长度
	单个数据集	链接数据集		
D1：Music Brainz/ BBC Music （MusicArtist/ MusicArtist）	255/0	255/255	33227/19653	6.8/5.3
D2：Music Brainz/DBpedia (MusicArtist/MusicalArtist)	0/55	255/55	33227/22135	6.8/5.7
D3：FOAF Profiles/Dbpedia (Person/ Person)	28/6	31/15	17062/36541	5.3/3.5
D4：Geonames/Dbpedia (#S. AIRP/Airport)	378/35	382/82	9768/16525	12.5/5.1
D5：FOAF Profiles/SIOC Site （OnlineAccount/UserAccount）	0/72	42/76	8812/5603	3.9/7.2
D6：SmanticWeb. org/ DBLP （Proceedings/ proceedings）	32/233	126/241	19720/26564	7.8/11.3

总体来看，由于使用不同的模式生成规则，建立模式级链接的关联数据集上的 RDF 数据项模式明显多于单个数据集。对于类似于 DBpedia 这种涵盖多领域、较为通用的数据集，所生成的数据项模式数量少于领域特色、专业性更加鲜明的特殊数据集；从产生的事务数量来看，由于 Dbpedia 包含的实例数量更加丰富，在数据项模式数据量相等的条件下，通用数据集中将产生规模更大的事务集合；事务平均长度反映了数据项模式中包含的谓词的数量，由于特定领域本体对相同概念的描述更丰富细致，因此事务长度略高于通用数据集。

(2)模式级链接对 RDF 数据项生成效率的影响

RDF 数据项模式的产生建立在对领域本体中的概念、属性和公理进行解析的基础之上，未涉及对海量实例数据的分析，因此不是影响本文所提方法效率的关键。RDF 数据项的生成以及数据项频繁度的计算是决定频繁模式挖掘效率的主要因素，如第 4 节所述。为了对第 4 节提出的实现方法的效率进行较为准确的评价，本研究使用 DBpedia 数据集，选择以“Country”为用户关注概念，产生 RDF 数据项模式，首先选择其中 5 个模式(平均包含 3 个谓词)，计算过程中不利用模式之间的包含关系，生成了一组不同规模的实例数据，再将所生成的 RDF 数据项转换成事务型数据的格式，在具有不同节点数的 Hadoop 集群上所消耗的时间如表 4 中 A 列所示。所消耗的时间分为两部分：获得 RDF 数据项的时间(等价于从 RDF 数据集中查询满足数据项模式的实例的时间)和将数据项装换成事务型数据格式的时间(表 4 中括号内将两部分时间用“＋”相连)。非常明显，由于涉及到相对复杂的查询，RDF 数据项的获取占用了绝大多数时间；事务型数据格式转换相当于对 RDF 数据项的重新排序，不需要进行 RDF 数据查询，在 MapReduce 框架下，其仅耗费相对较少的时间。

表 4　DBpedia 数据集上生成“Country”相关 RDF 数据项所耗费的时间(s)

数据集规模	4 个节点		9 个节点		14 个节点	
	A	B	A	B	A	B
121MB	249(244＋5)	137	231(227＋4)	124	205(202＋3)	103
224MB	297(288＋9)	172	275(266＋9)	146	237(229＋8)	117
510MB	416(399＋17)	236	389(374＋15)	189	351(338＋13)	152
1.3GB	833(796＋37)	419	812(778＋34)	366	773(741＋32)	304
2.4GB	1925 (1862＋63)	978	1883 (1824＋59)	849	1846 (1789＋57)	735

其次对所选的 5 个 RDF 数据项模式使用性质 1，即在 k-模式的查询结果的基础上查询 $k+1$ 模式的结果，生成“Country”相关 RDF 数据项所耗费时间，如表 4 中 B 列所示(不包含转换为事务型数据格式所耗费的时间)。从结果可以看出，性质 1 可明显减少数据处理时间，但每次需要以 k-模式的查询结果为待处理数据重新对 $k+1$ 模式的查询分配任务。由于两种方式得到的查询结果是一致的，因此两种情况下对获得的 RDF 数据项进行事务型数据格式转换所耗费的时间相同。

(3)模式级链接对关联规则数量的影响

本文以 DBpedia 和 Geonames 数据集为例来说明模式级链接对关联数据集合上关联规则的影响。仍然以“Country”为用户关注概念，分别在 DBpedia 和 Geonames 中选择 200000 个与“Country”有关的三元组，建立基于 RDF 数据项的事务型数据格式，当支持度取值在 0.5～0.8 之间时，所产

生的频繁模式数量和关联规则数量如表5所列。关联规则的分布情况呈现出明显的特征：DBpedia通用数据集上仅产生少量的关联规则，绝大多数的规则来自于领域特征突出的专业数据集，通过模式级链接而获得的关联规则也有相当可观的数量。根据扩展数据项模式生成规则，模式级链接关联数据集上的数据项模式可能会包含来自于同领域内不同本体的概念或属性，这使得RDF数据项所表达的事实是属于用户所关注的领域内的，而不受限于某个特定的数据集。这种形式的关联规则能更深刻地反映出隐藏在海量关联数据中的信息的价值。

表5 DBpedia和Geonames上以"Country"为核心概念的关联规则数量

支持度	频繁项集	规则数量				平均置信度
		总数	DBpedia	Geonames	模式级链接规则	
0.5	276572	1637846	983(0.06%)	1477337(90.2%)	159526(9.74%)	0.955
0.55	90277	532612	586(0.11%)	486275(91.3%)	45751(8.59%)	0.955
0.6	16648	60497	55(0.09%)	55778(92.2%)	4664(7.71%)	0.952
0.65	1205	5010	46(0.91%)	4474(89.3%)	490(9.79%)	0.942
0.7	299	922	15(1.66%)	770(83.5%)	137(14.84%)	0.933
0.75	53	127	10(7.9%)	102(80.6%)	15(11.5%)	0.925
0.8	3	6	0(0%)	6(100%)	0(0%)	0.924

结束语 本文主要研究了建立模式级链接对关联数据集上的数据挖掘过程的影响以及基于RDF数据模型的语义特征的关联规则挖掘过程。针对关联数据呈现出的大数据特性，通过建立模式级链接的方式对关联数据集合上的关联规则挖掘方法进行了深入研究，提出了基于RDF数据项模式的数据项生成机制，进而提出关联数据集上的关联规则挖掘方法。研究工作的核心是在深入分析关联数据自身结构特征的基础上，针对关联数据存在的问题提出一种实现关联数据模式级链接的方式，进而能够深层次利用RDF数据模型所具有的语义知识和推理能力，最终实现关联数据集上的有效的关联规则挖掘。通过本文方法所获取的关联规则容易理解，并且不受数据集边界的限制，能深层次反映特定领域内蕴含的规律。本文研究在实现RDF数据关联挖掘的各个阶段都从关联数据的特征出发，同时充分利用已经相对成熟的经典的关联规则挖掘技术。实验结果证明了模式级链接对关联数据集上的数据挖掘中所发挥的积极作用，同时对本文所提方法的有效性进行了验证。由于目前关于关联数据集上的数据挖掘结果仍相对较少，还缺少能广泛使用的测试集，因此难以准确地实现与同类方法的比较。进一步工作包括两方面：1)提高建立关联数据模式级链接的准确性和有效性，减少或避免人工的参与；2)深入研究经典关联规则挖掘方法作用于RDF数据集的优化机制，主要考虑如何利用RDF数据项模式和RDF数据项的特征来减少频繁模式发现过程中的计算量，从不同层面提升关联数据集上的数据挖掘性能。

参考文献

[1] Hausenblas M, Karnstedt M. Understanding Linked Open Data as a Web-Scale Database[C]// 2010 Second International Conference on Advances in Databases Knowledge and Data Applications (DBKDA). 2010:56-61

[2] Quboa Q K, Saraee M. A State-of-the-Art Survey on Semantic Web Mining[J]. Intelligent Information Management, 2013, 5(1):10-17

[3] Rettinger A, Losch U, Tresp V, et al. Mining the Semantic Web-Statistical Learning for Next Generation Knowledge Bases[J]. Data Mining and Knowledge Discovery, 2012, 24(3):613-662

[4] Nebot V, Berlanga R. Finding association rules in semantic web data[J]. Knowledge-Based Systems, 2012, 25(1):51-62

[5] Yazdi A S H, Kahani M. A novel model for mining association rules from semantic Web data[C]// 2014 Iranian Conference on Intelligent Systems (ICIS). 2014:1-4

[6] Abedjan F N Z. Context and Target Configurations for Mining RDF Data[C]// Proceedings of the 1st International Workshop on Search and Mining Entity-relationship Data (SMER'11). 2011

[7] Abedjan Z, Naumann F. Improving RDF Data Through Association Rule Mining[J]. Datenbank-Spektrum, 2013, 13(2):111-120

[8] Jiang C, Coenen F, Zito M. A Survey of Frequent Subgraph Mining Algorithms[J]. Knowledge Engineering Review, 2004, 28(1):75-105

[9] Wang Y, Ramon J. An efficiently computable support measure for frequent subgraph pattern mining[M]// Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 2012:362-377

[10] Narasimha R I V, Vyas O P. LiDDM: A Data Mining System for Linked Data[C]// Proceedings of the LDOW2011. Hyderabad, India, 2011

[11] Khan M A, Grimnes G A, Dengel A. Two pre-processing operators for improved learning from semantic Web data[C]// First Rapid Miner Community Meeting And Conference (RCOMM). 2010

[12] Kiefer A B C, Locher A. Adding data mining support to SPARQL via statistical relational learning[C]// Proceedings of the 5th European Semantic Web Conference on the Semantic Web: Research and Applications Methods (ESWC'08). 2008:478-492

[13] Husain M F, Khan L, Kantarcioglu M, et al. Data intensive query processing for Semantic Web data using Hadoop and MapReduce[M]. The University of Texas at Dallas, 2011

[14] Husain M F, McGlothlin J, Masud M M, et al. Heuristics-Based Query Processing for Large RDF Graphs Using Cloud Computing[J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(9):1312-1327

[15] Yuan P, Xie C, Jin H, et al. Dynamic and fast processing of queries on large-scale RDF data[J]. Knowledge and Information Systems, 2014, 41(2):311-334

[16] Ali L, Janson T, Schindelhauer C. Towards Load Balancing and Parallelizing of RDF Query Processing in P2P Based Distributed RDF Data Stores[C]// 2014 22nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). IEEE, 2014:307-311