基于命题逻辑的关联规则挖掘算法 L-Eclat

徐 卫 李晓粉 刘端阳

(浙江工业大学计算机科学与技术学院 杭州 310023)

摘 要 关联规则挖掘是数据挖掘领域非常重要的课题,在很多领域被广泛应用。关联规则挖掘算法都需要设置最小支持度和最小置信度。很多国内外学者研究的挖掘算法在这两方面都存在着一些问题,不仅需要大量的领域知识来设置合适的最小支持度,而且其结果集庞大、用户不容易理解。针对关联规则挖掘算法存在的问题,将命题逻辑融合到关联规则算法 Eclat 中,设计出了基于命题逻辑思想的挖掘算法 L-Eclat。实验结果表明,L-Eclat算法压缩了挖掘的规则集,减小了算法的时间消耗,且即使是非常小的支持度也可以得到高质量的关联规则,这在一定程度上解决了支持度设置的问题。

关键词 关联规则,命题逻辑,支持度,置信度

中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.12.038

Propositional Logic-based Association-rule Mining Algorithm L-Eclat

XU Wei LI Xiao-fen LIU Duan-yang

(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract Association rule mining is an important topic in the field of data mining, and it has been widely used in lots of practical applications. Generally, association rule mining algorithms have to set the minimal support threshold and the minimal confidence threshold. But it is hard for most mining algorithms to set these two values. Not only is tremendous related knowledge needed to select the support threshold, but also the mining results are too large and difficult to understand. To solve these problems, the idea of propositional logic was introduced into Eclat, which is one of the classical association rule mining algorithms. We proposed logic-based association rule mining algorithm called L-Eclat. Then, we compared L-Eclat with Eclat. The results show that L-Eclat can optimize and compress the result rule sets at certain degree, and it results in less time consumption and high-quality association rules. Furthermore, L-Eclat can run with a smaller support threshold, and it decreases the dependence on the support threshold and avoids spending much time on choosing a suitable support threshold.

Keywords Association rule, Propositional logic, Support threshold, Confidence threshold

1 概述

关联规则挖掘是数据挖掘领域中一个非常活跃的研究课题,可以广泛应用于很多领域。有效地发现、理解和运用关联规则是完成数据挖掘任务的重要手段,关联规则挖掘的研究具有重要的理论价值和现实意义[1]。关联规则挖掘,就是从数据库中找出事务属性之间的关系[2]。关联规则的挖掘过程主要分为两步:1)设置最小支持度来发现频繁项集;2)设置最小置信度来找出满足条件的规则[3]。关联规则挖掘算法在使用中主要存在两方面的问题:1)支持度的设置需要大量的领域知识或者不断的尝试探索;2)挖掘的结果集非常庞大,用户难以理解,且算法执行时间长、效率低。本文主要研究上述问题。

2 研究现状

为了解决关联规则挖掘中存在的问题,国内外学者对关联规则挖掘算法进行了深入的研究。在关联规则挖掘过程中会产生大量的频繁项集,为了解决该问题,崔亮等[4]提出了一种基于动态散列和事务压缩的关联规则挖掘算法,该算法能够动态地使用散列技术压缩频繁项集的规模,减少对数据库的扫描次数,提高算法的效率;谢志鹏等[5]结合概念格和关联规则,根据问题的实际情况对概念格的结构进行修改,提出了基于概念格的关联规则挖掘算法和渐进式的生成算法,该算法能够压缩规则集;李云等[6]提出了一种基于概念格的简洁关联规则的算法,该算法所挖掘到的规则能够满足用户的要求,并且规则集是完备的;欧阳继红等[7]在 FP-growth 算法的

到稿日期:2016-06-30 返修日期:2016-09-30 本文受浙江省自然科学基金(LY14F020018)资助。

徐 卫(1966-),男,硕士,高级工程师,主要研究方向为数据挖掘;李晓粉(1989-),女,硕士,主要研究方向为数据挖掘;刘端阳(1975-),男,博士,副教授,主要研究方向为数据挖掘、分布式计算,E-mail:ldy@zjut.edu.cn(通信作者)。

基础上提出了一种具有动态加权特性的挖掘算法，其权值具有动态可变性，可以剔除那些权重小且无意义的规则，从而压缩了规则集；段军等[8]提出了基于多支持度的挖掘关联规则算法 AM-WARMS，该算法可以设置多个支持度，在一定程度上解决了传统关联规则挖掘算法中支持度设置过小而导致的爆炸式增长问题；Li 等[9]提出了一种基于粗糙集的算法模型来对关联规则的重要性进行排序，从而得到高质量的关联规则；Hu 等[10]将分类算法和关联规则挖掘算法相融合，形成一种新的高效的挖掘框架，该框架提高了算法的时间效率，使得规则集得到压缩，更加易于理解。

以上所述算法都在一定程度上改进了关联规则挖掘算法，但它们都只是应用各种技术（如动态散列、概念格、动态加权法、多支持度、粗糙集理论和多算法融合等）来压缩规则集规模，从而提高算法的效率，然而每一种算法在支持度的设置问题上还是需要经验知识和多次试探，规则的完备性和可理解性也仍然存在问题。Sim 等[11]提出了一种完全不同的新思路来解决支持度和置信度问题，他们提出基于命题逻辑的模式挖掘算法，该算法不需要提前设定支持度阈值来发现频繁项集，减少了对领域知识的依赖，并且可以通过逻辑等价对一些规则进行筛选，保留了满足命题逻辑的规则，提高了结果集的置信度和可用性。本文将借鉴该思路，融合命题逻辑思想来研究支持度和置信度问题。

Eclat 算法[12-13]是经典的关联规则挖掘算法，不同于以往使用水平数据格式的数据库的算法，该算法使用垂直数据格式的数据库来挖掘频繁项集的挖掘算法，只需要扫描两次数据库，大大减少了挖掘规则所需要的时间。然而，Eclat 算法也存在支持度设置问题和规则冗余性问题。本文将命题逻辑思想融合到 Eclat 算法中，设计了一种基于命题逻辑的关联规则挖掘算法 L-Eclat，解决了 Eclat 算法存在的支持度设置问题、规则可理解性差以及规模庞大的问题。

3 Eclat 算法

3.1 Eclat 算法思想

Eclat 算法即等价类交换算法，使用垂直数据格式的数据集进行频繁项集的挖掘。垂直数据格式是类似于⟨*item*，*TID_set*⟩格式的数据集，其中 *item* 指项目的名称，*TID_set* 指含有 *item* 的所有事务的编号集合。与垂直数据格式相对应的是水平数据格式。水平数据格式指事务数据集是以下格式：⟨*TID*；*itemsets*⟩，*TID* 指事务的编号，*itemsets* 指事务所包含的项目集。

3.2 Eclat 算法描述

Eclat 算法采用垂直数据格式表示，在挖掘频繁项集的过程中首先对数据库进行第一次扫描，将水平格式的数据变换成垂直格式的数据。项集的支持度的计数就是 *TID_set* 集合中元素的总个数，因此在该过程中，候选集的产生和计数能够同时完成。从 $k=1$ 开始，根据先验知识，使用频繁 k 项集来构造候选的 $k+1$ 项集，构造的方法是通过将 k 频繁项集的 *TID_set* 相交来计算 $k+1$ 项集的 *TID_set*。一直重复该过程，直到不再有新的频繁项集产生。Eclat 算法的伪代码如算法 1 所示。

算法 1 Eclat 算法

输入：事务数据 D 和最小支持度阈值 MinS

输出：满足最小支持度阈值的频繁项集 L

```
1. 第一次扫描事务数据库 D，将水平格式的数据集转化成垂直格式的
   数据集；
2. 第二次扫描数据库得到频繁 1 项集，记作 L1，L=L∪L1；
3. Eclat(L1)：
4. {
5.    for all Xi∈L1 do{
6.       Xj∈L1，j<i do{
7.          通过交叉计算产生新的候选集 R=Xi∪Xj，
8.          TID_set(R)=TID_set(Xi)∩TID_set(Xj)；
9.          if (|TID_set(R)|≥MinS)
10.          {L=L∪R；
11.           Ti=Ti∪R；//Ti 开始时为空
12.           if (Ti≠∅) 调用 Eclat(Ti)；
13.           else exit；
14.          }
15.      }
16.   }
17. }
```

3.3 Eclat 算法的不足

Eclat 算法在产生频繁项集时使用了先验知识，能够压缩结果集，提高算法效率；同时在计算支持数时不需要重新扫描数据库，因为每一个 k 项集的 TID 本身就携带了支持度的完整信息。然而，算法还是存在以下不足。

(1)在支持度的设置上，需要大量的领域知识或者不断的尝试探索。

(2)所挖掘到的结果集庞大，用户难以理解这些结果。

(3)Eclat 算法在搜索策略上采取的是深度优先，因而不能应用先验知识对候选集剪枝，在数据量较小时对算法的影响不大，当数据量较大时，会增大搜索的空间，降低算法性能。

4 融合命题逻辑的 Eclat 算法

4.1 命题逻辑的相关理论

在哲学、逻辑学和语言学中，命题是一个能够表示一个判断陈述的语义，在数学中是一个表示判断一件事情的陈述句。逻辑指一种思维规律和客观规律。命题逻辑就是通过一些简单的逻辑运算将一些原子命题结合起来的公式和规则[14]。在逻辑中有两种重要的关系：蕴含关系和等价关系。

4.1.1 蕴含关系

蕴含在不同的学科中具有不同的含义。在命题逻辑中，蕴含主要用来描述两个命题之间的联系，主要使用推论的方式来对具有蕴含式关系的命题进行真假判断。每一个蕴含命题可以被认为已经符合逻辑规则，这些逻辑规则可以是逻辑等价或者语意蕴含，每一个蕴含命题都可以使用一个真值表来判断真假。例如，用 x 和 y 来表示两个不同的命题，并由这两个命题组成不同的蕴含命题，则这些蕴含命题的结论有可

能为真也有可能为假。x 和 y 可以组成以下 4 条不同的蕴含命题[11]：

1）$x\rightarrow y$

2）$\neg x\rightarrow y$

3）$\neg x\rightarrow y$

4）$\neg x\rightarrow\neg y$

从以上 4 个蕴含命题中可以发现，命题中都包含“→”，这是逻辑蕴含的标识，“¬”表示逻辑非运算。为了更好地理解这些符号，下面给出一个例子：

用 x 表示顾客购买了啤酒，y 表示顾客购买了尿布。

如果“顾客购买了啤酒”，那么“顾客也购买了尿布”，用蕴含式表示时为：$x\rightarrow y$；

如果“顾客购买了啤酒”，那么“顾客没有购买尿布”，用蕴含式表示时为：$x\rightarrow\neg y$。

在实际应用中，可以根据命题 x 和 y 的交集得到蕴含命题的真假。如 x 为真，且 y 为真，可以得到蕴含关系 $x\rightarrow y$ 为真。

4.1.2　等价关系

在蕴含关系中有一种特殊的关系——等价关系。蕴含关系能够用真值表来判断真假，等价关系同样也具有真值表。等价关系的命题表示为 $x\equiv y$，该命题为真的条件是当且仅当 $\neg(x\wedge y)$，“∧”指异或，即只有 x 和 y 的取值相同时，才为真。等价关系的真值表如表 1 所列。

表 1　等价关系的真值表

x	y	$x\equiv y$
T	T	T
T	F	F
F	T	F
F	F	T

等价关系能够使两个命题相关联，并且不依赖于用户的领域知识，即等价关系能够完全利用命题逻辑对规则判断真假，不需要考虑详细的领域知识。如此，将关联规则映射为等价关系是可行的。在将关联规则映射为等价关系时，能够依赖逻辑找到有趣的关联规则，该过程不需要依赖相关领域知识，挖掘到的规则具有很高的应用价值。

4.2　等价映射

在数据库中存在很多条记录，记录中包含很多项目；在一个记录中，某一个项目可能出现，也可能不出现。当某一项目出现在一条记录中时，映射为一个命题，且该命题为真；当一个项目未出现在一条记录中，映射为另一个命题，该命题为假[9]。将记录中的项目映射为命题 x 和 y，具体的映射如下：

1）项目 X 映射为 $x=T$，当且仅当项目 X 包含在记录中；

2）项目 X 映射为 $x=F$，当且仅当项目 X 未包含在记录中；

3）项目 Y 映射为 $y=T$，当且仅当项目 Y 包含在记录中；

4）项目 Y 映射为 $y=F$，当且仅当项目 Y 未包含在记录中。

同理，一条关联规则也可以映射为一个蕴含命题，蕴含命题的值也就是关联规则的值。假设一条关联规则包含两个项目 X 和 Y，项目 X 和 Y 能够被映射为 $x=T$ 和 $y=T$，当且仅当记录中同时包含 X 和 Y。

将关联规则映射为等价关系，必须满足一定的映射条件[11]，映射的具体条件如表 2 所列。

表 2　规则和关系的映射条件

等价关系	$x\equiv y$	$\neg x\equiv\neg y$
关联规则	$X\rightarrow Y$	$\neg X\rightarrow\neg Y$
关联规则的真假	将规则映射为等价关系需要满足的条件	
T	$X\rightarrow Y$	$\neg X\rightarrow\neg Y$
F	$X\rightarrow\neg Y$	$\neg X\rightarrow Y$
F	$\neg X\rightarrow Y$	$X\rightarrow\neg Y$
T	$\neg X\rightarrow\neg Y$	$X\rightarrow Y$

表 2 中 X 和 Y 是记录中的两个不同项目，关联规则 $X\rightarrow Y$ 有趣并且能映射为等价关系 $x\equiv y$，当且仅当：规则 $X\rightarrow Y$ 为真；规则 $X\rightarrow\neg Y$ 为假；规则 $\neg X\rightarrow Y$ 为假；规则 $\neg X\rightarrow\neg Y$ 为真。

当数据库中包含多条记录时，关联规则 $X\rightarrow Y$ 能映射为等价关系 $x\equiv y$，当且仅当：

$$Sup(XY)>Sup(X\rightarrow Y) \tag{1}$$

$$Sup(XY)>Sup(\neg XY) \tag{2}$$

$$Sup(\neg X\rightarrow Y)>Sup(X\rightarrow Y) \tag{3}$$

$$Sup(\neg X\rightarrow Y)>Sup(\neg XY) \tag{4}$$

这 4 个不等式同时满足。

在以上 4 个不等式中，$Sup(XY)$ 表示同时包含项目 X 和项目 Y 的记录数，$Sup(X\rightarrow Y)$ 表示包含项目 X 但不包含项目 Y 的记录数，$Sup(\neg XY)$ 表示不包含项目 X 但包含项目 Y 的记录数，$Sup(\neg X\rightarrow Y)$ 表示既不包含项目 X 也不包含项目 Y 的记录数。那么，$Sup(XY)>Sup(X\rightarrow Y)$ 表示 X 和 Y 同时出现的次数大于 X 出现但 Y 不出现的次数。根据映射条件，规则的列联表如表 3 所列。

表 3　规则的列联表

同时包含前后件的记录数		后件 Y	
		Y	$\neg Y$
前件 X	X	$Q_1=Sup(XY)$	$Q_1=Sup(X\rightarrow Y)$
	$\neg X$	$Q_1=Sup(\neg XY)$	$Q_1=Sup(\neg X\rightarrow Y)$

4.3　L-Eclat 算法设计

本节将命题逻辑的思想应用到关联规则挖掘过程中，在设置较小的支持度时能够压缩挖掘到的规则集，并过滤掉不合逻辑的规则，增强了挖掘结果的可理解性。这样，一方面，可以减小规则挖掘过程中对领域知识的依赖性，在一定程度上解决因支持度设置问题而带来的一些负面影响；另一方面，通过逻辑等价对规则进行过滤，保留了出现次数较少但有价值的规则，这样可以排除一些不合逻辑且有误导性的信息，有助于决策者做出更好的决策。

L-Eclat 算法的详细描述如算法 2 所示。

算法 2　L-Eclat 算法

输入：事务数据库 D 和最小支持度阈值 min_sup

输出：可信度高的压缩关联规则集

1. 第一次扫描数据库，将保存为水平格式的数据库转化为垂直格式的数据库。

2. 计算事务数据库中的每一项 i_m 的支持数，该支持数的大小就是 TID 的长度。将得到的支持度和最小支持度阈值 min_sup 进行比较，若支持度大于 min_sup，则将该项目添加到项集 L_1，L_1 初始为空。

3. 若项集 L_k 不为空，则取 L_k 中频繁项集的 TID 集的交来计算 k+1 项集的 TID 集。

4. 计算 k+1 项集的 TID 集中每个 TID 集的长度，也即该 k+1 项集的支持数，与最小支持度阈值 min_sup 进行比较，若项集出现的次数大于最小支持度数，则将该项集添加到频繁 k+1 项集 L_{k+1} 中。

5. 融合命题逻辑的思想。对于频繁 k+1 项集 L_{k+1} 的每一个项集 l_m，产生〈A,B〉形式的项目集，其中项目集 A 和 B 为 l_m 的子集，并且 $A \cup B = l_m$，$A \cap B = \varnothing$。

6. 对于每一个项目集 A 和 B，将项目集 A 看作是前件，B 看作是后件，计算 3 个项目集的支持数：$Q_1 = Sup(AB)$，$Q_2 = Sup(\neg AB)$，$Q_3 = Sup(A \neg B)$。在计算 Q_2 和 Q_3 时，可以使用以下两个公式：

$$Q_2 = (Sup(A) - Q_1)$$

$$Q_3 = (Sup(B) - Q_1)$$

7. 对项目集〈A,B〉进行验证，观察 $Q_1 > Q_2$，$Q_1 > Q_3$ 是否满足，若满足，则将 l_m 加入 L_{k+1} 中。

8. 对于所有的 L_k，只要 L_k 不为空，$k = k+1$，$L_{all} = L_{all} \cup L_k$，重复执行步骤 2—7；若 L_k 为空，继续向下执行。

9. 生成规则集。对于任意项目集 $l_i \subset L_{all}$，生成规则 $(A \rightarrow B)$，其中 $A \cup B = l_m$，$A \cap B = \varnothing$，计算规则的置信度：$conf(A \rightarrow B) = Sup(AB)/Sup(A)$ 和提升度：$lift(A \rightarrow B) = conf(A \rightarrow B)/Sup(B)$。

10. 输出所有的规则、支持度和置信度到文件中。

5 实验结果及分析

5.1 实验环境与数据

实验算法采用 Java 实现，在 Eclipse 上编译执行，操作系统为 Windows7。程序运行的环境为 Intel Core i3-330M 2.13 GHz CPU，2GB DDR3。实验选取真实数据集 MUSHROOM 作为数据挖掘的对象。MUSHROOM 数据集中记录了各种蘑菇的特征参数。MUSHROOM 数据集中主要包括 8124 条数据记录，每一条记录中都包括 23 个项目，该数据能够经常被用于关联规则的挖掘和分类过程。数据的详细描述如表 4 所列。

表 4 MUSHROOM 数据描述

数据集名称	MUSHROOM
事务记录数	8124
事务平均长度	23
最大事务长度	23
最小事务长度	23
项目数量	23

5.2 实验结果及分析

本节对融合命题逻辑的关联规则挖掘算法 L-Eclat 和 Eclat 进行了实验，实验选择最小支持度的变化范围为 0.2～0.5，记录生成规则的数目、规则的平均置信度和算法的执行时间，实验的结果如表 5 所列。从表 5 中可以发现，当支持度设置为 0.20 时，Eclat 算法得到的规则数是 762942，而 L-Eclat 算法得到的规则数是 1634，规则的平均置信度分别为 68.4%和 77.3%，详细情况如图 1 和图 2 所示。从图 1 中可以发现，当支持度设置得较小时，Eclat 算法挖掘到的规则数多于 L-Eclat 挖掘到的规则数；从图 2 中可以发现，L-Eclat 所挖掘到的规则的平均置信度大于 Eclat 算法所挖掘到的规则的置信度，这说明基于命题逻辑的 L-Eclat 算法能够压缩规则集，挖掘到的规则的可信度更高，即使在支持度设置得较小时也能够得到较好的规则集。在算法的运行效率方面（见图 3），当支持度从 0.20 变化到 0.50 时，L-Eclat 的运行时间短于 Eclat 的运行时间，这说明将命题逻辑引入 Eclat 中后，算法的运行时间缩短，效率得到提高。

表 5 Eclat 和 L-Eclat 的实验结果

算法	最小支持度	生成规则数目	平均置信度/%	运行时间/s
Eclat 算法	0.20	762942	68.4	86.3
	0.25	53744	67.8	12.3
	0.30	24640	70.3	8.3
	0.35	9078	72.4	6.5
	0.40	3812	74.3	5.4
	0.45	2098	76.1	4.6
	0.50	860	80.0	4.2
L-Eclat 算法	0.20	1634	77.3	41.4
	0.25	1458	77.0	7.1
	0.30	1396	77.2	6.8
	0.35	1230	78.3	5.2
	0.40	1108	78.6	5.2
	0.45	1026	78.9	4.6
	0.50	860	80.0	4.1

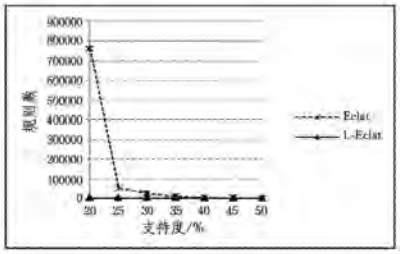

图 1 Eclat 和 L-Eclat 算法挖掘的规则数比较

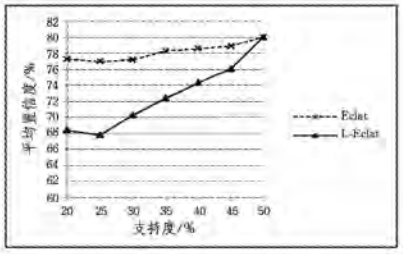

图 2 Eclat 和 L-Eclat 算法挖掘的规则平均置信度比较

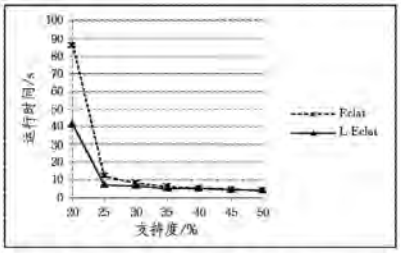

图 3 Eclat 和 L-Eclat 算法的运行时间比较

结束语 基于命题逻辑的关联规则挖掘算法 L-Ecalt 的时间效率比传统的挖掘算法 Eclat 的时间效率高，能够过滤到大量不符合逻辑的规则，同时还能压缩规则集，挖掘到的规则集的可信度更高，更加容易被理解，减小了对支持度阈值和

相关领域知识的依赖，即使在支持度设置得较小时，也不会产生大量的规则，并且挖掘到的规则的置信度更高。

参 考 文 献

[1] HAN J，KAMBER M. 数据挖掘概念与技术[M]. 范明，孟小峰，译. 北京：机械工业出版社，2006：1-27.

[2] RAKESH A，SRIKANT R. Fast Algorithms for Mining Association Rules[C]// Proceedings of International Conference on Very Large DataBases. Santiago，Chile：ACM Press，1994：21-30.

[3] 李锦泽，叶晓俊. 关联规则挖掘算法研究现状[C]// 计算机技术与应用进展——全国计算机技术与应用. 安徽：中国科学技术大学出版社，2007：9-14.

[4] CUI L，GUO J，WU L D. Algorithm for Mining Association Rules Based on Dynamic Hashing and Transaction Reduction [J]. Computer Science，2015，42(9)：41-44. (in Chinese)
崔亮，郭静，吴玲达. 一种基于动态散列和事务压缩的关联规则挖掘算法[J]. 计算机科学，2015，42(9)：41-44.

[5] XIE Z P，LIU Z T. Concept Lattice and Association Rule Discovery [J]. Journal of Computer Research & Development，2000，37(12)：1415-1421. (in Chinese)
谢志鹏，刘宗田. 概念格与关联规则发现[J]. 计算机研究与发展，2000，37(12)：1415-1421.

[6] LI Y，LI T，CAI J J，et al. Extracting Succinct Association Rules Based on Concept Lattice[J]. Journal of Nanjing University of Posts and Telecommunications (Natural Science)，2007，27(3)：44-47. (in Chinese)
李云，李拓，蔡俊杰，等. 基于概念格提取简洁关联规则[J]. 南京邮电大学学报(自然科学版)，2007，27(3)：44-47.

[7] OUYANG J H，WANG Z J，LIU D Y. An Improved Association Rule Algorithm with Dynamically Weighted Characteristic[J]. Journal of Jilin University (Science Edition)，2005，43(3)：314-319. (in Chinese)
欧阳继红，王仲佳，刘大有. 具有动态加权特性的关联规则算法[J]. 吉林大学学报(理学版)，2005，43(3)：314-319.

[8] DUAN J，DAI J F. Algorithm of Mining Weighted Association Rules Based on Multiple Supports[J]. Journal of Tianjin University，2006，39(1)：114-118. (in Chinese)
段军，戴居丰. 基于多支持度的挖掘加权关联规则算法[J]. 天津大学学报，2006，39(1)：114-118.

[9] LI J，CERCONE N. A Rough Set Based Model to Rank the Importance of Association Rules [C]// Proceedings of Rough Sets，Fuzzy Sets，Data Mining，and Granular Computing. Berlin Heidelberg：Springer Press，2005：109-118.

[10] HU K，LU Y，ZHOU L，et al. Integrating Classification and Association Rule Mining：A Concept Lattice Framework[C]// Proceedings of New Directions in Rough Sets，Data Mining，and Granular-Soft Computing. Berlin Heidelberg：Springer Press，2003：443-447.

[11] SIM A，INDRAWAN M，ZUTSHI S，et al. Logic-Based Pattern Discovery [J]. IEEE Transactions on Knowledge and Data Engineering，2010，22(6)：798-811.

[12] ZAKI M. Scalable Algorithms for Association Mining[J]. IEEE Transactions on Knowledge and Data Engineering，2000，12(3)：372-390.

[13] ZAKI M，GOUDA K. Fast Vertical Mining using Diffsets [C]// Proceedings of International Conference on Knowledge Discovery and Data Mining. Washington DC：ACM Press，2003：326-335.

[14] CHEN C H，LAN G C，HONG T P，et al. Mining High Coherent Association Rules with Consideration of Support Measure [J]. Expert Systems with Applications，2013，40(16)：6531-6537.

[15] AN J R，WANG H P，ZHANG L B，et al. A Compression Matrix Algorithm for Mining Association Rules Based on Mapreduce[J]. Journal of Chongqing University of Technology(Natural Science)，2016，30(2)：95-100. (in Chinese)
安建瑞，王海鹏，张龙波，等. 一种基于 MapReduce 的压缩矩阵关联规则挖掘算法[J]. 重庆理工大学学报(自然科学版)，2016，30(2)：95-100.

(上接第 193 页)

[13] FRANCO-SALVADOR M，CRUZ F L，TROYANO J A，et al. Cross-domain polarity classification using a knowledge-enhanced meta-classifier [J]. Knowledge-Based Systems，2015，86：46-56.

[14] ZHANG Z F，MIAO D Q，NIE J Y，et al. Sentiment uncertainty measure and classification of negative sentences [J]. Journal of Computer Research and Development，2015，52(8)：1806-1816. (in Chinese)
张志飞，苗夺谦，聂建云，等. 否定句的情感不确定性度量及分类[J]. 计算机研究与发展，2015，52(8)：1806-1816.

[15] ZHANG Z F，MIAO D Q，YUE X D，et al. Sentiment analysis with words of strong semantic fuzziness [J]. Journal of Chinese Information Processing，2015，29(2)：68-78. (in Chinese)
张志飞，苗夺谦，岳晓冬，等. 强语义模糊性词语的情感分析[J]. 中文信息学报，2015，29(2)：68-78.

[16] DAS S，CHEN M. Yahoo! for Amazon：Extracting market sentiment from stock message boards [C]// Proceedings of the Asia Pacific Finance Association Annual Conference. 2001：43.

[17] LI S S，LEE S Y M，CHEN Y，et al. Sentiment classification and polarity shifting [C] // Proceedings of the 23rd International Conference on Computational Linguistics. ACL，2010：635-643.

[18] YAO Y Y. Three-way decisions with probabilistic rough sets [J]. Information Sciences，2010，180(3)：341-353.

[19] YAO Y Y，ZHAO Y. Attribute reduction in decision-theoretic rough set models[J]. Information Sciences，2008，178(17)：3356-3373.

[20] QIU G，LIU B，BU J，et al. Expanding domain sentiment lexicon through double propagation [C]// Proceedings of the 21st International Joint Conference on Artificial Intelligence. Morgan Kaufmann，2009：1199-1204.

[21] XIA R，XU F，ZONG C Q，et al. Dual sentiment analysis：Considering two sides of one review [J]. IEEE Transactions on Knowledge and Data Engineering，2015，27(8)：2120-2133.