文章编号: 1001-9081(2007)03-0561-04

模糊时序关联规则挖掘

崔晓军[1,2], 薛永生[2]
(1. 襄樊职业技术学院 信息技术系, 湖北 襄樊 441050; 2. 厦门大学 计算机科学系, 福建 厦门 361005)
(cxjxhy@163.com)

摘 要: 借助模糊概念和模糊运算, 对时间区间的描述很容易实现。对于指定的日历模式, 不同的时间区间可根据它们的隶属度具有不同的权重。在模糊日历代数基础上, 结合增量挖掘和累进计数的思想, 提出了一种基于模糊日历的模糊时序关联规则挖掘方法。理论分析和实验结果均表明, 该算法是高效可行的。

关键词: 关联规则; 模糊数据挖掘; 模糊集; 模糊日历代数
中图分类号: TP311.13 文献标识码: A

Mining fuzzy temporal association rules

CUI Xiao-jun[1,2], XUE Yong-sheng[2]
(1. Department of Information Technology, Xiangfan Vocational and Technical College, Xiangfan Hubei 441050, China;
2. Department of Computer Science, Xiamen University, Xiamen Fujian 361005, China)

Abstract: With the help of fuzzy calendar algebra and fuzzy operators, it is easy to describe desired temporal requirements. Time intervals may have different weights according to their membership functions. Integrated with the idea of progression and increment, an algorithm combined with incremental mining and progressive counting called BFCTAR was proposed to mine fuzzy temporal association rules based on fuzzy calendar algebra. Theoretic analysis and experimental results indicate that this algorithm is efficient and feasible.

Key words: association rule; fuzzy data mining; fuzzy set; fuzzy calendar algebra

0 引言

近来, 时序关联规则挖掘已经越来越引起研究者的关注, 文献[1]提出周期性关联规则的挖掘, 文献[2]提出循环关联规则挖掘, 文献[3,4]提出日历关联规则挖掘。由于周期性和循环模式建立在单一的时间粒度上, 而日历模式建立在多时间粒度上, 这与实际生活中的年、月、日, 时、分、秒等多粒度时间表示更吻合, 因此基于日历的时序关联规则挖掘更有实用价值。

文献[4]的方法允许用户指定感兴趣的时间区间, 文献[3,5]中的算法可以发现在指定日历模式下所有的时序关联规则, 然而上述方法的前提均是基于精确的时间区间描述, 但现实中用户很难进行精确描述, 因此模糊时序关联规则挖掘更有现实意义。

在文献[6]提出的模糊日历代数的基础上, 综合文献[7,8]提出的累进和文献[9]提出的增量处理方法, 本文提出了基于模糊日历的模糊时序关联规则挖掘方法。

1 相关概念

1.1 模糊日历代数

自从美国科学家扎德(L. A. Zadeh)1965年提出模糊集合的概念, 模糊集理论已广泛应用在各个领域。借助模糊集理论, 文献[10]提出了模糊日历代数的概念, 利用模糊代数, 用户可以定义多时间粒度的复杂的日历, 对于指定的日历模式, 不同的时间区间可根据它们的隶属度具有不同的权重。

日历是一个时间区间的结构化的集合。然而, 对用户来讲, 精确地描述他们期望的日历模式是不太可能的, 借助模糊概念和模糊运算, 日历的描述就容易实现了。

日历的构造离不开层次的时间粒度, 对于每个时间粒度, 模糊集可以用于描述该时间粒度内所有的时间区间的贡献。对于某个时间粒度的模糊描述(例如一年的中间、一个月的最初、周末)就形成一个基本的模糊日历。

定义 1 基本模糊日历。给定一个时间粒度 U, 基本模糊日历 A 通过隶属函数 μ_A 描述了该时间粒度下所有的时间区间的模糊表示。形式地, $\forall T_i \in U$, $\mu_A: U \rightarrow [0,1]$, 函数值 $\mu_A(T_i)$ 代表时间区间 T_i 对于基本模糊日历 A 的隶属度。

图 1给出了基本模糊日历的实例。通常选用梯形和三角形分布为模糊日历的隶属函数, 但用户也可根据需要自己任意地指定隶属函数的图形, 还可以使用“很”、“多”、“少”等形容词来进行模糊描述。对于模糊日历来说, 直观上容易看出, 重要的时间区间隶属度也较大, 对日历的贡献也较大, 如对图 1中的模糊日历“周初”, 周 1和周 2最重要, 隶属度为 1, 周 3次之, 隶属度为 0.5, 周 4以后则没有关系, 隶属度为 0。模糊日历同样可以用于精确的时间区间描述, 如图 2所示。对于时序关系(如“之前”、“之后”等), 模糊日历也可通过阶梯状的隶属函数进行描述, 如图 3所示。

收稿日期: 2006-09-12 基金项目: 福建省自然科学基金资助项目(A0310008); 福建省高新技术研究开放计划重点项目(2003H043)
作者简介: 崔晓军(1972-), 男, 湖北襄樊人, 副教授, 硕士研究生, 主要研究方向: 数据仓库、数据挖掘; 薛永生(1946-), 男, 福建福清人, 教授, 主要研究方向: 数据库理论与应用、分布式数据库、数据仓库、数据挖掘、网络技术。

在现实世界中，复杂的时间表达(如年末月中，但不在周初)是很常见的、有用的描述。在基本模糊日历的基础上，借助与(and)、或(or)、非(not)、异或(xor)和减(sub)五种运算可以很方便地构造任意复杂的模糊日历。

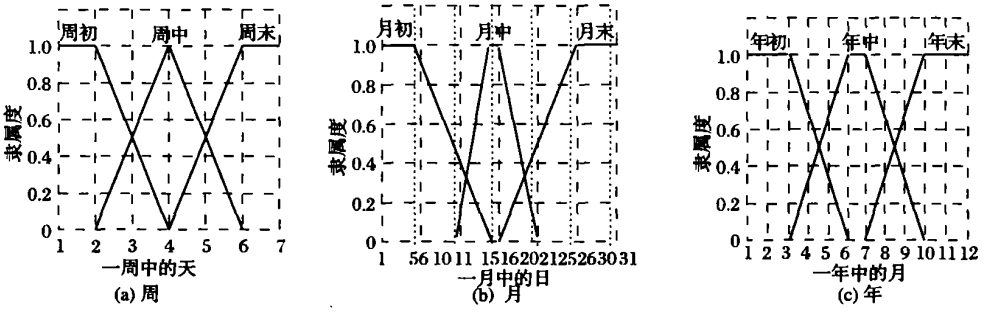

图 1 不同时间粒度下的基本模糊日历

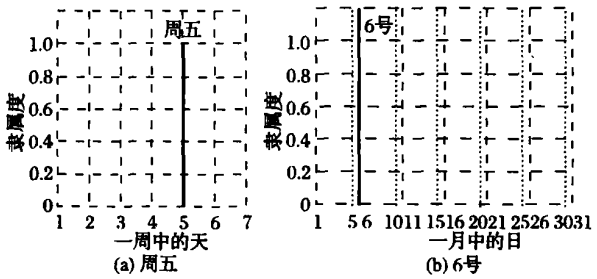

图 2 精确的时间区间描述

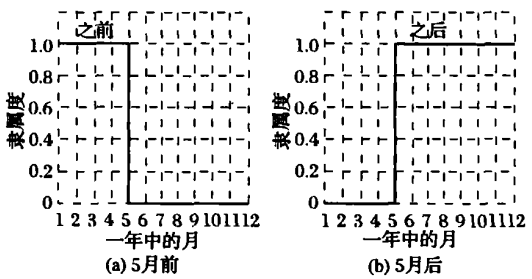

图 3 时序关系的描述

定义 2 模糊日历。可以递归地定义如下：1)一个基本模糊日历是一个模糊日历；2)令 A 和 B 是两个模糊日历，则 A and B、A or B、not A、A xor B 和 A sub B 运算的结果也是模糊日历。

这些运算符的语义描述如下：

令 A 和 B 是两个模糊日历，分别具有隶属度 μ_A 和 μ_B，则：

1) A and B 表示为 $A \wedge B$，运算结果的隶属函数 $\mu_{A\wedge B} \equiv \mu_A \times \mu_B$；

2) A or B 表示为 $A \vee B$，运算结果的隶属函数 $\mu_{A\vee B} \equiv \mu_A + \mu_B - \mu_A \times \mu_B$；

3) not A 表示为 $\neg A$，运算结果的隶属函数 $\mu_{\neg A} \equiv 1 - \mu_A$；

4) A xor B 表示为 $A \oplus B$，运算结果的隶属函数 $\mu_{A\oplus B} \equiv \mu_A \times (1-\mu_B)^2 + \mu_B \times (1-\mu_B)^2 + \mu_A \times \mu_B \times (1 - \mu_A \times \mu_B)$；

5) A sub B 表示为 $A - B$，运算结果的隶属函数 $\mu_{A-B} \equiv \mu_A - \mu_A \times \mu_B$。

在这里用代数的和、积来定义 or 和 and 运算，而没有用 max、min 主要原因在于 max 或 min 在某些场合不适用，例如考虑两个模糊集的 and 运算时，希望大的模糊集对结果也有大的影响，如果采用 min 运算就会导致结果没有影响了。同理，max 运算在处理 or 运算时也不适合。

对于基本模糊日历“周初，周中，周末，月初，月中，月末，年初，年中，年末”，可以简记为 wb、wm、we、mb、mm、me、yb、ym、ye，其隶属函数分别表示为 μ_{wb}、μ_{wm}、μ_{we}、μ_{mb}、μ_{mm}、μ_{me}、μ_{yb}、μ_{ym}、μ_{ye}。则一个复杂的模糊日历可以分解为以上基本模糊日历的组合。

例 1 一个复杂模糊日历 $C1$：月中且年末，或者，周末且年初。这个日历的隶属函数 μ_{C1} 可以表示为：

$$\begin{aligned}\mu_{C1} &= \mu_{(mm\wedge ye)\vee(we\wedge yb)}\\ &= \mu_{(mm\wedge ye)} + \mu_{(we\wedge yb)} - \mu_{(mm\wedge ye)} \times \mu_{(we\wedge yb)}\\ &= \mu_{mm}\times\mu_{ye} + \mu_{we}\times\mu_{yb} - \mu_{mm}\times\mu_{ye}\times\mu_{we}\times\mu_{yb}\end{aligned}$$

例 2 一个复杂模糊日历 $C2$：不在年末，并且，在年中但不在月末。这个日历的隶属函数 μ_{C2} 可以表示为：

$$\begin{aligned}\mu_{C2} &= \mu_{(\neg ym)\wedge(ym\wedge\neg me)}\\ &= \mu_{(\neg ym)}\times\mu_{(ym\wedge\neg me)}\\ &= (1-\mu_{ym})\times\mu_{ym}\times(1-\mu_{me})\end{aligned}$$

例 3 一个复杂模糊日历 $C3$：在月初或月末，但不是同时。这个日历的隶属函数 μ_{C3} 可以表示为：

$$\begin{aligned}\mu_{C3} &= \mu_{(mb\oplus me)}\\ &= \mu_{mb}\times(1-\mu_{me})^2 + \mu_{me}\times(1-\mu_{mb})^2 +\\ &\quad \mu_{mb}\times\mu_{me}\times(1-\mu_{mb}\times\mu_{me})\end{aligned}$$

某个具体的时间 T 相对于模糊日历 A 的隶属度(时间 T 在 A 内)记为 $\sigma_A(T)$，利用以上运算符可以方便地计算出来。

例 4 时间 T = 2006/08/20(周日)对于模糊日历 $C1$ 的隶属度 $\sigma_{C1}(T)$，根据图 1 的隶属度函数可以计算如下：

$$\begin{aligned}\mu_{C1} &= \mu_{mm}\times\mu_{ye} + \mu_{we}\times\mu_{yb} - \mu_{mm}\times\mu_{ye}\times\mu_{we}\times\mu_{yb}\\ &= \mu_{mm}(20)\times\mu_{ye}(8) + \mu_{we}(7)\times\mu_{yb}(8) -\\ &\quad \mu_{mm}(20)\times\mu_{ye}(8)\times\mu_{we}(7)\times\mu_{yb}(8)\\ &= 0.2\times 0.33 + 1.0\times 0 - 0.2\times 0.33\times 1.0\times 0\\ &= 0.066\end{aligned}$$

1.2 模糊时序关联规则

设 $I = \{i_1, i_2, \cdots, i_m\}$ 是数据项的集合，任务相关的数据 D 是数据库事务集合，D 中的每个事务 T 都具有一个时间戳 TID，且 $T \subseteq I$ 时间信息 t_i 指示事务发生的时间。假定事务数据库 D 被划分为 n 个集合 $D_1, D_2, \cdots, D_n$，每个 D_i 包含所有在最小时间粒度决定的时间段 T_i 内发生的事务(如最小粒度为天，则每个 D_i 包含一天内发生的事务)。模糊时序关联规则挖掘的任务就是发现在模糊日历模式给定的时间区间内有趣的关联规则。这些有趣的规则的加权支持度和加权置信度应该大于用户指定的最小支持度和最小置信度阈值。

设 FC 为用户指定的模糊日历，w_i 为相应于分区 D_i 的时间段 T_i 的隶属度，w_i 的计算过程如例 4。对于给定的项集 $A \subseteq I$，事务 t 包含 A 当且仅当 $A \subseteq T$，设 $|D_i(I)|$ 表示分区 D_i 中包含项集 A 的事务数，则项集 A 在 D 中的加权计数记为 $\sigma_D(A)$。

$$\sigma_D(A) = \sum_{i=1}^{n}(|D_k(A)|\times w_k) \quad (1)$$

项集 A 是频繁的,当且仅当其加权计数 $\sigma_D(A)$ 满足支持度阈值 $minsupp$,即:

$$\sigma_D(A) \geqslant [\sum_{i=1}^{n}(|D_k|\times w_k)]\times minsupp\% \quad (2)$$

定义 3　模糊关联规则。是给定模糊日历 FC 下的形如 $X \overset{FC}{\Rightarrow} Y$ 的蕴含式,其中 $X \subseteq Y$, $Y \subseteq I$ 且 $X \cap Y = \varnothing$。模糊关联规则 $X \overset{FC}{\Rightarrow} Y$ 具有加权支持度 $s\%$,当且仅当 $\sigma_D(X \cup Y) = [\sum_{i=1}^{n}(|P_k|\times w_k)]\times s\%$, $X \overset{FC}{\Rightarrow} Y$ 具有加权置信度 $c\%$,当且仅当 $\sigma_D(X \cup Y)/\sigma_D(X) = c\%$。

2　理论依据及基本思想

综合文献[7, 8]提出的累进和文献[9]提出的增量处理方法,本文提出了基于模糊日历的模糊时序关联规则挖掘方法,其基本思想如下:首先,把事务数据库划分为 n 个分区(D_1, D_2, …, D_n),每个分区只包括由模糊日历中的最小时间粒度所确定的基本时间段(如最小时间粒度为天,则基本时间段为一天)的事务,对每个分区 D_i 计算其权重(隶属度)w_i,对于分区 D_i,定义其加权的计数阈值 m_i 为:

$$m_i = w_i \times |D_i| \times minsupp\% \quad (3)$$

其中,$|D_i|$ 是分区 D_i 内的事务数,$minsupp$ 是用户指定的支持度阈值。

分区的累积加权的计数阈值 M_i 定义为:

$$\begin{cases} M_{ii} = m_i \\ M_{ij} = M_{i(j-1)} + m_j \end{cases} \quad (1 \leqslant i \leqslant j \leqslant n) \quad (4)$$

首先生成候选频繁 2项集(C_2),然后通过 C_2 生成所有的候选频繁项集(C),最后通过一次扫描数据库 D 生成频繁项集(L),关联规则即可产生。

候选频繁 2项集(C_2)的生成采用累进的方法[8],一个 2项集 I 对于分区 D_j 是部分频繁的,当且仅当存在一个分区 D_i($1 \leqslant i \leqslant j \leqslant n$)使得 I 在分区 D_i, D_{i+1}, …, D_j 中的累进加权计数 $WC_{ij}(I)$ 大于等于加权计数阈值 M_{ij}。$WC_{ij}(I)$ 的定义如下:

$$\begin{cases} WC_{ii}(I) = \sigma_{D_i}(I) \\ WC_{ij}(I) = WC_{i(j-1)}(I) + \sigma_{D_j}(I) \end{cases} \quad (5)$$

其中 $\sigma_{D_i}(I)$ 是项集 I 在分区 D_i 中的加权计数,定义如下:

$$\sigma_{D_j}(I) = w_i \times |D_i(I)| \quad (6)$$

从第一个分区 D_1 开始依次处理每个分区,最初候选频繁 2项集(C_2)为空。对于分区 D_j 中的每个 2项集,存在两种情况:1)如果 $I \notin C_2$ 且 $\sigma_{D_j}(I) \geqslant m_j$,则将 I 加入到 C_2 中,并且将 D_j 和 $\sigma_{D_j}(I)$ 分别记录为 I 的起始分区和累进加权计数;2)如果 $I \in C_2$,假设已记录的 I 的起始分区和累进加权计数分别为 D_i 和 V,则若 $V + \sigma_{D_j}(I) \geqslant M_{ij}$,则仍将 I 保留在 C_2 中,只将其累进加权计数改为 $V + \sigma_{D_j}(I)$;若 $V + \sigma_{D_j}(I) < M_{ij}$,则将 I 从 C_2 中删除,同时删除关于项集 I 的所有记录信息。很明显,当处理完最后一个分区 D_n 后,保留在 C_2 中的 2项集一定是关于分区 D_n 部分频繁的,证明见文献[8]。同时,由于在处理过程中不断删除不满足条件的项集,C_2 也很接近于 L_2,这使得 C_2 成为一个好的候选集。

然后利用 C_2 来连接生成其他的候选 k项集($k \geqslant 3$):$C_{k+1} = C_k \infty C_k$, ∞ 代表文献[11]中的连接操作。对于任意的 $R \in C_{k+1}$, R 的所有 k 项子集一定包含在 C_k 中(Apriori性质),因此,候选项集的集合 C 是所有候选 k项集($k \geqslant 2$)的并集,即 $C = \bigcup_{k \geqslant 2} C_k$。

最后通过扫描数据库 D 一次,由 C 生成频繁项集 L。对于 C 中的每个项集 I 已经记录了 $\sigma_D(I)$。如果 $\sigma_D(I) \geqslant M_{1n}$(数据库 D 的加权计数阈值),则 I 即是 D 的频繁项集,否则 I 即是非频繁的。数据库 D 的频繁项集 $L = \bigcup_{k \geqslant 2} L_k$,通过频繁项集 L 可方便地得到模糊时序关联规则,挖掘任务完成。

3　算法描述

算法 1　BFCTAR(基于模糊日历的时序关联规则挖掘算法)

输入:事务数据库 D;最小支持度阈值 minsupp;最小置信度阈值 minconf;模糊日历 c

输出:模糊日历下的频繁项集 L

算法过程:

```
将 D 划分为 n个不相交的分区 D1,D2,…,Dn,每个分区包含最小时间粒度所确定的一个时间段内的记录
计算每个分区的权重(隶属度)wi
计算每个分区的加权计数阈值 mi:mi = wi×|Di|×minsupp
M11 = m1
FOR i=1 to n
    FOR j=2 to n
        Mij = M(i,j-1) + mj
    END FOR
END FOR
C2 = ∅
FOR each 2项集 I∈ D1
    IF σD1(I) ≥ m1
        C2 = C2 ∪ {I}
        Ds(I) = D1                    //Ds(I):项集 I起始分区
        V(I) = σD1(I)                 //V(I):I的累积加权计数
    END IF
END FOR
FOR j=2 to n
    FOR each 2-项集 I∈ Dj
        {IF I∉ C2 and σDj(I) ≥ mj
            C2 = C2 ∪ {I}
            Ds(I) = Dj
            V(I) = σDj(I)
        END IF
        IF I∈ C2
            IF V(I) + σDj(I) ≥ MDs(I)
                V(I) = V(I) + σDj(I)
            ELSE
                C2 = C2 - {I}
            END IF
        END IF}
    END FOR
END FOR
C = ∅
WHILE C2 ≠ ∅ and k ≥ 2
    Ck+1 = Ck ∞ Ck
    C = Ck ∪ Ck+1
END WHILE
L = ∅
FOR each I∈ C
    σD(I) = Σ(i=1..n) σDi(I)
    IF σD(I) ≥ M1n
        L = L ∪ {I}
    END IF
```

ENDFOR
RETURN L_k;

4 实验设计与分析

4.1 实验设计

在支持度阈值不同的情况下，对 BCTAR 算法和文献[10]提出的 Apriori+算法从执行时间、I/O 代价、生成的平均候选项集数这几个指标进行比较。实验数据采用文献[11]中介绍的方法生成的人造数据：T10.I4.D100K，其中 T 代表事务平均大小，I代表可能的最大频繁项集的项数，D 代表事务数据库的事务数，以 K 为单位。实验采用的模糊日历如例 1(月中且年末，或者，周末且年初)。事务数据库划分为 100个分区，每个分区中的 1 000条记录对应于该模糊日历的最小的时间粒度(天)。实验环境中的硬件平台为联想开天 M6200(P4 3.0G，1G RAM)，操作系统 Windows 2003 Sever 数据库系统 SQL Server 2000，编程环境为 Jbuilder 2005。

4.2 实验结果与分析

BFCTAR算法与 Apriori+算法的比较结果如图 4～图 6 所示。可看出，基于模糊日历的 BFCTAR 算法的执行效率优于 Apriori+算法，特别是在支持度阈值较小时；其 I/O 代价也明显小于 Apriori+算法；其产生的候选项集数接近于频繁项集数，明显小于 Apriori+算法所产生的候选项集数。其原因主要在于 BFCTAR算法在生成候选项集的过程中采用累进的方法，每扫描一个分区即记录候选项集的相应信息，减少了扫描数据库的次数，并且在处理过程中不断删除不满足条件的项集，所以生成的候选项集数也很接近于频繁项集。

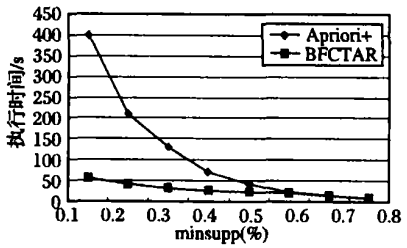

图 4 BFCTAR算法和 Apriori+算法执行时间比较

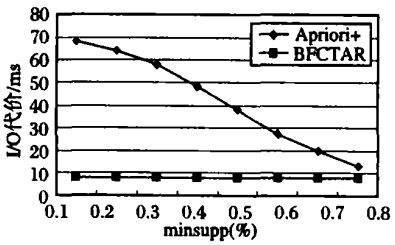

图 5 BFCTAR算法和 Apriori+算法 I/O 代价比较

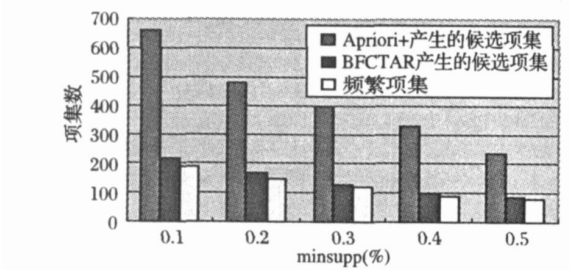

图 6 BFCTAR算法和 Apriori+算法生成的候选项集数比较

参考文献：

[1] HAN J, DONG G, YIN Y. Efficient Mining of Partial Periodic Patterns in Time Series Database[A]. Proceedings of the International Conference on Data Engineering[C]. 1999. 106－115.

[2] OZDEN B, RAMASWAMY S, SUKBERSCHATZ A. Cyclic Association Rules[A]. Proceedings of the 15th International Conference on Data Engineering[C]. 1998. 412－421.

[3] LI Y, NING P, WANG XS, et al. Discovering Calendar-based Temporal Association Rules[A]. Data and Knowledge Engineering[J]. 2003, 44(2): 193－218.

[4] RAMASWAMY S, MAHAJAN S, SILBERSCHATZ A. On the Discovery of Interesting Patterns in Association Rules[A]. Proceedings of the International Very Large Database Conference[C]. 1998. 368－379.

[5] LEE W J, JIANG JY, LEE SJ. An Efficient Algorithm to Discover Calendar-based Temporal Association Rules[A]. Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics[C]. 2004. 3122－3127.

[6] LEE W J, LEE SJ. Fuzzy Calendar Algebra and Its Applications to Data Mining[A]. Proceedings of 11th International Symposium on Temporal Representation and Reasoning[C]. 2004. 71－78.

[7] LEE CH, LIN CR, CHEN MS. Sliding-window filtering: An efficient algorithm for incremental mining[A]. Proceedings of ACM 10th International Conference on Information Knowledge Management[C]. Atlanta, GA, 2001. 263－270.

[8] LEE CH, OU JC, CHEN MS. Progressive weighted mining: An efficient method for time-constraint mining[A]. Proceedings of 7th Pacific Asia Conference on Knowledge Discovery Data Mining[C]. Seoul, Korea, 2003. 449－460.

[9] CHEUNG DW, LEE SD, KAO B. A general incremental technique for maintaining discovered association rules[A]. Proceedings of 5th International Conference on Database Systems: Advanced Application[C]. Melbourne, Australia, 1997. 185－194.

[10] ALE JM, ROSSI GH. An Approach to Discovering Temporal Association Rules[A]. Proceedings of the 2000 ACM Symposium on Applied Computing[C]. 2000. 294－300.

[11] AGRAWAL R, SRIKANT R. Fast algorithms for mining association rules[A]. Proceedings of 1994 International Conference on Very Large Database Bases: VLDB'94[C]. 1994.