模糊 Horn 子句规则挖掘算法研究

刘东波[1,2] 卢正鼎[1]

(华中科技大学计算机科学与技术学院 武汉 430074)[1] (中国电子设备系统工程研究所 北京 100141)[2]

摘 要 模糊关联规则可以用自然语言来表达人类知识,受到数据挖掘与知识发现研究人员的广泛关注。但是,目前大多数模糊关联规则挖掘方法仍然基于经典关联规则的支持度和可信度测度。从模糊蕴涵的观点出发,定义了模糊 Horn 子句规则、支持度、蕴涵强度以及相关概念,提出了模糊 Horn 子句规则挖掘算法。该算法可以分解为 3 个步骤。首先,将定量数据库转换为模糊数据库。其次,挖掘模糊数据库中所有支持度不小于指定最小支持度阈值的频繁项目集。一旦得到了所有频繁项目集,就可以用一种直接的方法生成所有蕴涵强度不小于指定最小蕴涵强度阈值的模糊 Horn 子句规则。

关键词 模糊关联规则,模糊 Horn 子句规则,支持度,蕴涵强度,定量数据库,模糊数据库

中图法分类号 TP182 **文献标识码** A

Research on Algorithms for Mining Fuzzy Horn Clause Rules

LIU Dong-bo[1,2] LU Zheng-ding[1]

(College of Computer Science & Technology, Huazhong University of Science & Technology, Wuhan 430074, China)[1]

(Institute of China Electronic System Engineering, Beijing 100141, China)[2]

Abstract Fuzzy association rules can be used to represent human knowledge in terms of natural language, and has attracted a growing amount of attention from the communities of Data Mining and Knowledge Discovery. However, so far, most approaches of mining fuzzy association rules are based on the measures of support and confidence for classical association rules. From the viewpoint of fuzzy implications, fuzzy Horn clause rules, degree of support, implication strength and some related concepts were defined, and an algorithm was proposed for mining fuzzy Horn clause rules. This algorithm can be decomposed into three subprocess. First of all, a quantitative database is transformed into a fuzzy database. Secondly, all frequent itemsets in the fuzzy database that are contained in a sufficient number of transactions above the minimum support threshold are identified. Once all frequent itemsets are obtained, the desired fuzzy Horn clause rules above the minimum implication strength threshold can be generated in a straightforward manner.

Keywords Fuzzy association rules, Fuzzy horn clause rules, Degree of support, Implication strength, Quantitative databases, Fuzzy databases

1 引言

关联规则挖掘是数据挖掘与知识发现(DMKD)领域的重要研究方向。Agrawal, Imeilinski 和 Swami 于 1993 年首先提出在大型数据库中挖掘关联规则,并给出了第一个关联规则挖掘算法 AIS[1],它通过多次循环来发现数据库中的频繁项目集(Frequent Itemsets),进而发现数据库中不同项目之间的关联关系,为管理人员确定营销策略、提高决策水平提供支持。AIS 算法的思路非常清晰,但执行效率很低。经典 Apriori 算法是 Agrawal 和 Srikant 提出的基于频繁项目集的递归方法[2],它将关联规则挖掘算法分解为两个步骤:(1)找出所有支持度(Degree of Support)不小于指定最小支持度阈值的频繁项目集;(2)利用频繁项目集生成可信度(Degree of Confidence)不小于指定最小可信度阈值的关联规则。

模糊关联规则可以用自然语言来表达人类知识,近年来受到 DMKD 研究人员的普遍关注。但是,目前大多数模糊关联规则挖掘算法与经典 Apriori 算法相似,仍然沿用支持度和可信度测度来挖掘数据库中的模糊关联规则[3-12]。事实上,模糊关联规则可以有不同的解释,而且不同的解释对规则挖掘方法有很大影响。Hüllermeier 指出,在模糊关联规则挖掘中直接采用可信度会引发逻辑问题[13]。于是,有些学者提出了基于蕴涵的模糊关联规则挖掘方法,例如 Hüllermeier 和 Beringer 研究了不同类型基于蕴涵的模糊关联规则及其挖掘方法[13,14]。Dubois, Hüllermeier 和 Prade 探讨了模糊关联规则支持度和可信度的多种计算方法[15],包括引入模糊蕴涵算子。陈国清、闫鹏等引入了蕴涵度概念,提出了基于蕴涵的模糊关联规则挖掘算法[16,17],并指出通过适当限制 t-模算子和模糊蕴涵算子,可以提高数据挖掘效率。高雅、马琳和戴齐提

到稿日期:2010-10-24 返修日期:2011-02-23

刘东波 高级工程师,主要研究方向为模糊逻辑、数据库、数据挖掘与知识发现、信息系统集成等;卢正鼎 教授,主要研究方向为数据库、数据挖掘、信息系统集成、计算机安全等。

出的模糊关联规则挖掘方法[18]不限制 t-模算子和模糊蕴涵算子的选择，而通过删除冗余候选规则提高挖掘效率。本文作者也从模糊逻辑的观点出发，初步研究了基于蕴涵的模糊逻辑规则发现方法[19]。

传统逻辑程序设计理论采用的是一阶谓词逻辑的 Horn 子句规则集合[20]。模糊 Horn 子句逻辑是对传统 Horn 子句逻辑的拓展[21,22]，它提供了一种模糊知识表示及其推理方法。但如何挖掘模糊 Horn 子句规则，特别是怎样确定规则的蕴涵强度，是制约模糊 Horn 子句逻辑应用的"瓶颈"问题。模糊关联规则挖掘为解决这一知识获取问题提供了可以借鉴的思路和方法。事实上，模糊 Horn 子句规则可以看成一种特殊的模糊关联规则。本文首先定义模糊 Horn 子句规则以及支持度、蕴涵强度等相关概念，然后提出模糊 Horn 子句规则挖掘算法，最后分析算法的复杂度。

2 模糊 Horn 子句规则及其相关概念

传统 Horn 子句规则是形如

$$A\leftarrow B_1\wedge B_2\wedge\cdots\wedge B_n, n\geqslant 0 \tag{1}$$

的逻辑蕴涵式。其中，规则体 $B_1\wedge B_2\wedge\cdots\wedge B_n$ 称为前提，规则头 A 称为结论[20]。

模糊 Horn 子句规则[21,22]与传统 Horn 子句规则类似，可描述为

$$A\leftarrow(f)-B_1\wedge B_2\wedge\cdots\wedge B_n, n\geqslant 0 \tag{2}$$

其中，规则体 $B_1\wedge B_2\wedge\cdots\wedge B_n$ 称为前提，规则头 A 称为结论；$f\in(0,1]$称为蕴涵强度(Implication Strength)，反映的是从前提到结论的真值传播。假设每个前提 B_i 的真值为 $t(B_i)$，$i=1,2,\cdots,n$，$B_1\wedge B_2\wedge\cdots\wedge B_n$ 的真值为 t。当 $n>0$ 时，按照模糊集合论的 max-min 原则，$t=\min\{t(B_i)\mid i=1,2,\cdots,n\}$；当 $n=0$ 时，令 $t=1$。于是，结论 A 的真值为$t(A)=f\otimes t$。这里$\otimes$是一个称为"软乘"的模糊蕴涵算子(Fuzzy Implication Operator)。当 $f=1$，且 $A,B_1,B_2,\cdots,B_n$ 的真值仅取 0 或 1 时，模糊 Horn 子句规则可以转化为传统 Horn 子句规则。

关于模糊蕴涵算子可参阅文献[23,24]，本文在不失一般性的前提下选用 $x\otimes y=\min\{1,1-x+y\}$。

假设 $I=\{i_1,i_2,\cdots,i_n\}$是定量数据库 D 中由 n 个不同项目(Items)组成的项目集(Itemsets)，其中 $i_k(k=1,2,\cdots,n)$的值域是 Ω_k，且 Ω_k 与若干模糊属性相关联。利用每个模糊属性相应的隶属函数，可以把定量数据库 D 转换为模糊数据库 $\tilde{D}$，其中 $\tilde{D}$ 的项目集为 $\tilde{I}=\{\tilde{i}_{11},\tilde{i}_{12},\cdots,\tilde{i}_{1p},\tilde{i}_{21},\tilde{i}_{22},\cdots,\tilde{i}_{2q},\cdots,\tilde{i}_{n1},\tilde{i}_{n2},\cdots,\tilde{i}_{nr}\}$，而 $\tilde{I}$ 中所有项目的值域为$[0,1]$。

首先定义与模糊数据库 $\tilde{D}$ 相关的模糊 Horn 子句规则。

定义 1 令 $\tilde{I}$ 是模糊数据库 $\tilde{D}$ 的项目集，与 $\tilde{D}$ 相关的模糊 Horn 子句规则是形如 $\tilde{A}\leftarrow(f)-\tilde{B}_1\wedge\tilde{B}_2\wedge\cdots\wedge\tilde{B}_n$ 的逻辑表达式，其中 $\tilde{A},\tilde{B}_1,\tilde{B}_2,\cdots,\tilde{B}_n\in\tilde{I}$，$\tilde{B}_1\wedge\tilde{B}_2\wedge\cdots\wedge\tilde{B}_n$ 称为前提，$\tilde{A}$ 称为结论，$f\in(0,1]$称为蕴涵强度。

下面定义模糊 Horn 子句规则的两个重要测度：支持度和蕴涵强度。

定义 2 令 $\tilde{I}$ 是模糊数据库 $\tilde{D}$ 的项目集，对于任意给定的项目子集 $\tilde{S}=\{\tilde{i}_1,\tilde{i}_2,\cdots,\tilde{i}_m\}\subseteq\tilde{I}$，$\tilde{S}$ 在 $\tilde{D}$ 的第 k 条记录中的支持度为

$$SUPP_k(\tilde{S})=\min\{\mu_{1k}(\tilde{i}_1),\mu_{2k}(\tilde{i}_2),\cdots,\mu_{mk}(\tilde{i}_k)\} \tag{3}$$

式中，$\mu_{1k}(\tilde{i}_1),\mu_{2k}(\tilde{i}_2),\cdots,\mu_{mk}(\tilde{i}_k)$分别是 $\tilde{i}_1,\tilde{i}_2,\cdots,\tilde{i}_m$ 在第 k 个记录中的模糊属性值。

定义 3 令 $\tilde{I}$ 是模糊数据库 $\tilde{D}$ 的项目集，对于任意给定的项目子集 $\tilde{S}\subseteq\tilde{I}$，$\tilde{S}$ 在 $\tilde{D}$ 中的支持度为

$$SUPP(\tilde{S})=\sum_{k=1}^{|\tilde{D}|}SUPP_k(\tilde{S})/|\tilde{D}| \tag{4}$$

式中，$|\tilde{D}|$表示模糊数据库 $\tilde{D}$ 中的记录数。

定义 4 令 $\tilde{I}$ 是模糊数据库 $\tilde{D}$ 的项目集，对于任意的 $\tilde{A}$，$\tilde{B}_1,\tilde{B}_2,\cdots,\tilde{B}_n\in\tilde{I}$，前提为 $\tilde{B}_1\wedge\tilde{B}_2\wedge\cdots\wedge\tilde{B}_n$ 且结论为 $\tilde{A}$ 的模糊 Horn 子句规则在 $\tilde{D}$ 中的支持度为

$$SUPP=\sum_{k=1}^{|\tilde{D}|}SUPP_k(\{\tilde{A},\tilde{B}_1,\tilde{B}_2,\cdots,\tilde{B}_n\})/|\tilde{D}| \tag{5}$$

式中，$|\tilde{D}|$表示模糊数据库 $\tilde{D}$ 中的记录数。

定义 5 令 $\tilde{I}$ 是模糊数据库 $\tilde{D}$ 的项目集，对于任意的 $\tilde{A}$，$\tilde{B}_1,\tilde{B}_2,\cdots,\tilde{B}_n\in\tilde{I}$，前提为 $\tilde{B}_1\wedge\tilde{B}_2\wedge\cdots\wedge\tilde{B}_n$ 且结论为 $\tilde{A}$ 的模糊 Horn 子句规则在 $\tilde{D}$ 的第 k 个记录中的蕴涵强度为

$$IMP_k=SUPP_k(\{\tilde{A}\})\otimes SUPP_k(\{\tilde{B}_1,\tilde{B}_2,\cdots,\tilde{B}_n\}) \tag{6}$$

式中，$SUPP_k(\{\tilde{A}\})=\mu_k(\tilde{A})$是 $\tilde{A}$ 在第 k 个记录中的支持度，$SUPP_k(\{\tilde{B}_1,\tilde{B}_2,\cdots,\tilde{B}_n\})=\min\{\mu_{1k}(\tilde{B}_1),\mu_{2k}(\tilde{B}_2),\cdots,\mu_{mk}(\tilde{B}_k)\}$是$\{\tilde{B}_1,\tilde{B}_2,\cdots,\tilde{B}_n\}$在第 k 个记录中的支持度，$\otimes$是模糊蕴涵算子[21-24]。

由于本文选用了模糊蕴涵算子 $x\otimes y=\min\{1,1-x+y\}$，所以

$$IMP_k=\min\{1,1-\mu_k(\tilde{A})+\min\{\mu_{1k}(\tilde{B}_1),\mu_{2k}(\tilde{B}_2),\cdots,\mu_{mk}(\tilde{B}_k)\}\}\} \tag{7}$$

定义 6 令 $\tilde{I}$ 是模糊数据库 $\tilde{D}$ 的项目集，对于任意的 $\tilde{A}$，$\tilde{B}_1,\tilde{B}_2,\cdots,\tilde{B}_n\in\tilde{I}$，前提为 $\tilde{B}_1\wedge\tilde{B}_2\wedge\cdots\wedge\tilde{B}_n$ 且结论为 $\tilde{A}$ 的模糊 Horn 子句规则在 $\tilde{D}$ 中的蕴涵强度为

$$IMP=\sum_{k=1}^{|\tilde{D}|}IMP_k/|\tilde{D}| \tag{8}$$

式中，$|\tilde{D}|$表示模糊数据库 $\tilde{D}$ 中的记录数，IMP_k 是上述模糊 Horn 子句规则在 $\tilde{D}$ 的第 k 个记录中的蕴涵强度。

定义 7 令 α 和 β 分别为最小支持度阈值和最小蕴涵强度阈值，如果前提为 $\tilde{B}_1\wedge\tilde{B}_2\wedge\cdots\wedge\tilde{B}_n$ 且结论为 $\tilde{A}$ 的候选规则满足支持度 $SUPP\geqslant\alpha$ 且蕴涵强度 $IMP\geqslant\beta$ 的条件，则称 $\tilde{A}\leftarrow(f)-\tilde{B}_1\wedge\tilde{B}_2\wedge\cdots\wedge\tilde{B}_n$为强模糊 Horn 子句规则。其中，$f=IMP$。

3 模糊 Horn 子句规则挖掘算法

模糊 Horn 子句规则挖掘算法可分解为 3 个处理进程：

(1)执行算法 1，将定量数据库 D 转换为值域在$[0,1]$上的模糊数据库 $\tilde{D}$；

(2)执行算法 2，找出模糊数据库 $\tilde{D}$ 中所有支持度不小于指定的最小支持度阈值 α 的频繁项目集；

(3)执行算法 3，以频繁项目集为基础，生成候选模糊 Horn 子句规则集，进而找出所有蕴涵强度不小于指定的最小蕴涵强度阈值 β 的强模糊 Horn 子句规则。

算法 1 DB_Trans

输入：定量数据库 D，其项目集为 I

输出：模糊数据库 $\tilde{D}$，其项目集为 $\tilde{I}$

算法：

```
begin
for all t∈D do //t 是数据库 D 中的记录
for all i_k∈I do //i_k(k=1,2,…,n)是项目集 I 的元素
```

```
    for all 与 i_k 相关的模糊属性 ĩ_kj 的隶属函数 μ_ĩkj(i_k) do
        计算 ĩ_kj 的隶属度 μ // ĩ_kj∈Ĩ, μ∈[0,1]
        insert μ into D̃
    end for
  end for
end for
return D̃
end
```

算法 2 是一种与经典 Apriori 算法类似的算法，它通过扫描模糊数据库 $\tilde{D}$ 发现所有支持度不小于指定最小支持度阈值 α 的频繁项目集。该算法与经典 Apriori 算法的区别主要有以下 3 点：(1)$\tilde{D}$ 中所有元素的值域为[0,1]；(2)与定量数据库 D 的同一属性相关联的模糊项目集不再进行连接操作；(3)将剪枝策略 $SUPP(\tilde{A}\tilde{B}\tilde{C})\geqslant SUPP(\tilde{A}\tilde{B})+SUPP(\tilde{A}\tilde{C})-SUPP(\tilde{A})$($\tilde{A},\tilde{B},\tilde{C}$ 均为 $\tilde{D}$ 中的项目)应用到算法当中，以减少候选项目集的数量。

算法 2 Frequent_Itemsets_Gen

```
输入：模糊数据库 D̃，最小支持度阈值 α
输出：频繁项目集 L={L_1,L_2,…,L_k}
算法：
begin
L_1=1 元频繁项目集 // L_k 表示 k 元频繁项目集
for {k=2;L_{k-1}≠Φ;k++} do //生成 k 元候选项目集 C_k
    C_k=candidates_gen(L_{k-1})
    ∀C̃∈C_k
    // 利用 SUPP(ÃB̃C̃)≥SUPP(ÃB̃)+SUPP(ÃC̃)-SUPP(Ã) 剪枝
    if ∃Ã⊆C̃, B̃⊆C̃, SUPP(Ã)+SUPP(B̃)-SUPP(Ã∩B̃)≥α
    then
        C̃ to L_k, delete C̃ from C_k
    end if
      for all t∈D̃ do // t 是 D̃ 中的记录
        for all C̃∈C_k do
          计算 t 对 C̃ 的支持度 SUPP_t(C̃) // 根据定义 2
          计算所有 SUPP_t(C̃) 的累加和 Σ
        end for
        // 根据定义 3 计算 C̃ 在模糊数据库 D̃ 中的支持度
          SUPP(C̃)=Σ/|D̃|
      end for
        // 生成 k 元频繁项目集
        L_k=L_k+{C̃∈C_k|SUPP(C̃)≥α}
end for
return L={L_1,L_2,…,L_k} // 生成所有频繁项目集
end
```

在算法 2 中，$candidates\_gen(L_{k-1})$ 过程与经典 Apriori 算法的 $Apriori\_gen(L_{k-1})$ 类似，主要包括两个步骤：

(1)连接。为了得到 k 元频繁项目集 L_k，首先通过($k-1$)元频繁项目集 L_{k-1} 的连接操作生成 k 元候选项目集 C_k。假定 L_1 和 L_2 是 L_{k-1} 中的频繁项目集，$L_i[j]$ 表示 L_i 的第 j 项。如果 $L_1[1]=L_2[1]\wedge L_1[2]=L_2[2]\wedge\cdots\wedge L_1[k-2]=L_2[k-2]\wedge L_1[k-1]<L_2[k-1]$，则做连接操作 $L_1\bowtie L_2$，连接条件是 L_1 和 L_2 的前($k-2$)项相同，连接结果为 $L_1[1]L_1[2]\cdots L_1[k-1]L_2[k-1]$。

(2)剪枝。连接之后的结果 C_k 是 L_k 的超集，它的元素可能不是频繁项目集，这时就需要扫描模糊数据库 $\tilde{D}$，计算 C_k 中每个候选项目集的基数，以确定 L_k。利用“频繁项目集的任意子集都是频繁项目集”这一性质可以对 C_k 进行剪枝，把子集不在 L_{k-1} 中的候选项目集从 C_k 中删除。

```
Procedure candidates_gen(L_{k-1})
//L_{k-1}是(k-1)元频繁项目集
  for all L_1∈L_{k-1} do
    for all L_2∈L_{k-1} do
      if L_1[1]=L_2[1]∧L_1[2]=L_2[2]∧…∧L_1[k-2]=L_2[k-2]∧L_1[k-1]<L_2[k-1] then {
        C̃=L_1 ⋈ L_2 // ⋈ 是连接运算符；
                     //C̃ 是候选项目集
        if has_infrequent_subset(C̃,L_{k-1}) then
          delete C̃ // 剪枝操作
        else
            add C̃ to C_k
        }
return C_k
Procedure has_infrequent_subset(C̃,L_{k-1})
// C̃ 是 k 元候选项目集；L_{k-1}是(k-1)元频繁项目集
for all C̃ 的(k-1)元子集 S̃ do
    if S̃∉L_{k-1} then return true
return false
```

算法 3 以频繁项目集为基础，可以挖掘出所有蕴涵强度不小于最小蕴涵强度阈值 β 的强模糊 Horn 子句规则。

算法 3 Mining_Fuzzy_Horn_Clause_Rules

```
输入：频繁项目集 L={L_1,L_2,…,L_k}，最小蕴涵强度阈值 β
输出：强模糊 Horn 子句规则集 Γ
算法：
begin
for all l̃∈L_k do
  if ∀l̃-{Ã}={B̃_1,B̃_2,…,B̃_n} then
  // 计算候选规则 Ã←(f)-B̃_1∧B̃_2∧…∧B̃_n 的蕴涵强度 f
    f=implication_strength(Ã,B̃_1∧B̃_2∧…∧B̃_n)
  end if
  if f≥β then
      add Ã←(f)-B̃_1∧B̃_2∧…∧B̃_n to Γ
  end if
end for
return Γ
end
```

在算法 3 中调用了计算规则蕴涵强度的过程 $implication\_strength(\tilde{A},\tilde{B}_1\wedge\tilde{B}_2\wedge\cdots\wedge\tilde{B}_n)$。

```
Procedure implication_strength(Ã,B̃_1∧B̃_2∧…∧B̃_n)
    for all t∈D̃ do // t 是 D̃ 中的记录
      // 根据定义 5 计算候选规则在 D̃ 的记录 t 中的蕴涵强度
      IMP_t=min{1,1-μ_t(Ã)+min{μ_1t(B̃_1),μ_2t(B̃_2),…,μ_mt(B̃_t)})}
  计算所有 IMP_t 的累加和 Σ
    end for
      // 根据定义 6 计算候选规则在 D̃ 中的蕴涵强度
  IMP=Σ/|D̃|
return IMP
```

4 算法复杂度分析

模糊 Horn 子句规则挖掘算法分为 3 个处理进程，分别

通过算法1、算法2和算法3实现。

算法1通过一次扫描数据库，将定量数据库 D 转换为值域在[0,1]上的模糊数据库 $\tilde{D}$，其计算复杂度为 $O(|\tilde{D}|\times|\tilde{I}|)$，其中 $|\tilde{D}|$ 是模糊数据库 $\tilde{D}$ 中包含的记录数，$|\tilde{I}|$ 是模糊数据库 $\tilde{D}$ 的属性集 $\tilde{I}$ 包含的元素数。

算法2用来挖掘模糊数据库 $\tilde{D}$ 中所有支持度不小于最小支持度阈值的频繁项目集。该算法与经典 Apriori 算法类似，在执行过程中需多次扫描数据库，其计算复杂度是 $O(|\tilde{D}|^k)$，其中 $|\tilde{D}|$ 是模糊数据库 $\tilde{D}$ 中包含的记录数，k 是最大频繁项目集的长度。需要说明的是，为了减少候选项目集的数量，提高频繁项目集挖掘效率，经典 Apriori 算法利用性质“频繁项目集的任意子集都是频繁项目集”进行了剪枝操作。在此基础上，算法2又利用了如下剪枝策略：$SUPP(\tilde{A}\tilde{B}\tilde{C})\geqslant SUPP(\tilde{A}\tilde{B})+SUPP(\tilde{A}\tilde{C})-SUPP(\tilde{A})$，进一步减少了候选项目集的数量。

算法3以频繁项目集为基础生成候选规则集，进而挖掘出所有蕴涵强度不小于最小蕴涵强度阈值的强模糊 Horn 子句规则。该算法通过选择适当的模糊蕴涵算子，将数据库扫描次数减少到 $k-1$ 次，其计算复杂度为 $O(|\tilde{D}|\times|\tilde{I}|\times(k-1))$，其中 k 是最大频繁项目集的长度。

综合上述，由于通常 $|\tilde{I}|$ 远远小于 $|\tilde{D}|$，因此模糊 Horn 子句规则挖掘算法总的计算复杂度约为 $O(|\tilde{D}|^k)$。

结束语 本文从模糊蕴涵的观点出发，提出了模糊 Horn 子句规则挖掘方法，为解决模糊 Horn 子句逻辑应用的知识获取“瓶颈”问题提供了一条有效的途径。该算法结合了模糊蕴涵概念和 Apriori 挖掘算法，首先执行算法1，将定量数据库转换为模糊数据库。然后执行算法2，挖掘模糊数据库中所有支持度不小于指定最小支持度阈值的频繁项目集。为了进一步减少候选项目集的数量，提高频繁项目集挖掘效率，算法2利用了新的剪枝策略 $SUPP(\tilde{A}\tilde{B}\tilde{C})\geqslant SUPP(\tilde{A}\tilde{B})+SUPP(\tilde{A}\tilde{C})-SUPP(\tilde{A})$。最后执行算法3，以频繁项目集为基础生成候选规则集，进而挖掘出所有蕴涵强度不小于指定最小蕴涵强度阈值的模糊 Horn 子句规则。算法3通过选择适当的模糊蕴涵算子，减少了数据库扫描次数，提高了关联规则生成效率。

参 考 文 献

[1] Agrawal R, Imeilinski T, Swami A. Mining Association Rules Between Sets of Items in Large Databases[C]// Proceedings of the 1993 ACM SIGMOD. 1993:207-216

[2] Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules in Large Databases[C]// Proceedings of the 20th International Conference on Very Large Data Bases. 1994:487-499

[3] Cubero J C, Medina J M, Pons O, et al. Rules Discovery in Fuzzy Relational Databases[C]// Proceedings of the 3rd International Symposium on Uncertainty Modelling and Analysis. 1995:414-419

[4] Chien B C, Lin Z L, Hong T P. An Efficient Clustering Algorithm for Mining Fuzzy Quantitative Association Rules[C]// Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference. 2001:1306-1311

[5] Kaya M, Alhajj R. Genetic Algorithm Based Framework for Mining Fuzzy Associations Rules[J]. Fuzzy Sets and Systems, 2005,152(3):587-601

[6] Subramanyam R B V, Goswami A. Mining Fuzzy Quantitative Association Rules[J]. Expert Systems, 2006,23(4):212-225

[7] 李波，施国兴，王新. 一种挖掘广义模糊关联规则的方法[J]. 云南民族大学学报：自然科学版，2007,16(3):259-262

[8] Berberoglu T, Kaya M. Hiding Fuzzy Association Rules in Quantitative Data[C]// Proceedings of the 3rd International Conference on Grid and Pervasive Computing Workshop. IEEE Computer Society, 2008:387-392

[9] Chen Zuo-liang, Chen Guo-qing. Building an Association Classifier Based on Fuzzy Association Rules[J]. Computational Intelligence Systems, 2008,1(3):262-273

[10] Sheibani R, Ebrahimzadeh A. An Algorithm For Mining Fuzzy Association Rules[C]// Proceedings of the International Multi Conference of Engineers and Computer Scientists. Hong Kong: Newswood Limited, 2008:486-490

[11] Gupta M, Joshi R C. Privacy Preserving Fuzzy Association Rules Hiding in Quantitative Data[J]. Computer Theory and Engineering, 2009,1(4):1793-8201

[12] 霍纬纲，邵秀丽. 基于 TD-FP-growth 的模糊关联规则挖掘算法[J]. 控制与决策，2009,24(10):1504-1508

[13] Hüllermeier E. Implication-based fuzzy association rules[C]// Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery. 2001:241-252

[14] Hüllermeier E, Beringer J. Mining implication-based fuzzy association rules in databases[C]// Intelligent Systems for Information Processing: From Representation to Applications. Elsevier, 2003:137-147

[15] Dubois D, Hüllermeier E, Prade H. A note on quality measures for fuzzy association rules[C]// Proceedings of the 10th International Fuzzy Systems Association World Congress on Fuzzy Sets and Systems. 2003:346-353

[16] Chen Guo-qing, Yan Peng, Kerre E E. Computationally efficient mining for fuzzy implication-based association rules in quantitative databases[J]. General Systems, 2004,33(2/3):163-182

[17] 闫鹏，陈国清. 发现基于蕴涵的模糊关联规则[J]. 模糊系统与数学，2004,18(z1):279-283

[18] 高雅，马琳，戴齐. 模糊关联规则的挖掘算法[J]. 西南交通大学学报，2005,40(1):26-29

[19] 刘东波，卢正鼎. 数据库中的模糊逻辑规则发现[J]. 计算机工程与应用，2008,44(26):147-153

[20] Kowalski R A. Predicate Logic as a Programming Language[C]// Proceedings of the 1974 IFIP Congress. North Holland, Amsterdam, 1974:569-574

[21] 刘东波，卢正鼎. 模糊 Horn 子句逻辑形式系统[J]. 模糊系统与数学，2007,21(2):30-39

[22] Liu Dong-bo, Lu Zheng-ding. The Theory of Fuzzy Logic Programming[C]// Proceedings of Second International Conference of Fuzzy Information and Engineering (ICFIE). Advances in Soft Computing 40. Springer-Verlag, 2007:534-542

[23] Ruan D, Kerre E E. Fuzzy implication operators and generalized fuzzy method of cases[J]. Fuzzy Sets and Systems, 1993,54(1):23-38

[24] Bui C C, Le C N. Some Fuzzy Operators with Threshold and Application to Fuzzy Association Rules in Data Mining[J]. Advances in Fuzzy Mathematics, 2010,5(3):245-262