维普资讯 http://www.cqvip.com

基于 Apriori 算法的多循环关联规则挖掘综述*)

袁军鹏　朱东华
(北京理工大学管理与经济学院　北京100081)

摘　要　介绍了基于 Apriori 算法的关联规则挖掘的研究状况，对一些典型采掘算法进行了分析和评价，总结和归纳了国内外学者对 Apriori 的改进，展望了关联规则挖掘的未来研究方向。
关键词　数据挖掘，关联规则，频繁项集，循环算法

A Survey of Association Rules of Apriori Algorithms

YUAN Jun-Peng　ZHU Dong-Hua
(School of Management & Economics, Beijing Institute of Technology, Beijing 100081)

Abstract　Association rules are one of the most researched areas of data mining and have recently received much attention from the database community. They have proven to be quite useful in the marketing and retail communities as well as other more diverse fields. In this paper we provide an overview of association rule research.
Keywords　Data mining, Association rules, Frequent item-sets , Sequential algorithms

数据挖掘中一个最重要的问题是从交易数据库中发现关联规则(Association Rule)[1~4]。关联规则是表示数据库中一组互不相属对象之间某种关联关系的规则。文[5]于1993年首先提出了关联规则挖掘的思想，并讨论了从大型超市的顾客交易数据库中挖掘关联规则的问题。文[6]提出了 Apriori 和 Apriori-Tid 算法，奠定了布尔型关联规则挖掘的基础。

1　关联规则的基本模型

关联规则挖掘可用以下的形式描述[5,6,31]：

设 $T=\{t_1,t_2,\cdots,t_n\}$是一个历史交易数据库，t_i 表示 T 的第 i 条记录。交易数据库的字段包含两大部分。第一部分是不用来标志交易内容的，一般不是布尔值，如记录号(TID)、交易时间、交易地点等。这部分字段记为 A，$A=\{a_1,a_2,\cdots,a_r\}$。第二部分是用来标志交易内容的，字段的值均为布尔值，用于反映在某一次交易中，顾客是否购买了某个产品。这部分字段记为 I，$I=\{i_1,i_2,\cdots,i_m\}$其中的元素称为项(item)。$t_i[i_k]$表示项 i_k 在记录 t_i 上的值；$t_i[a_k]$表示字段 a_k 在记录 t_i 上的值。我们用来区分每一条交易。

设 $X=\{x_1,x_2,\cdots,x_p\}$和 $Y=\{y_1,y_2,\cdots,y_q\}$均是 I 的子集，并且 $X\cap Y=\phi$。如果 $X\subseteq t_i$，那么就称交易 t_i 包含 X。一个关联规则是形如 $X\Rightarrow Y$ 的蕴含式。规则 $X\Rightarrow Y$ 在交易数据库 T 中的支持度(support)是交易集中包含 X 和 Y 的交易数与所有交易数 n 之比，记为 $Supp(X\cup Y)$，即：

$$Supp(X\cup Y)=\frac{\sum_{t_i\in T}\prod_{z_k\in X\cup Y}t_i[z_k]}{n} \quad (1)$$

规则 $X\Rightarrow Y$ 在交易集中的可信度(confidence)是指包含 X 和 Y 的交易数与仅包含 X 的交易数之比，也就是 $X\cup Y$ 的支持度与 X 的支持度之比，记为 $conf(X\Rightarrow Y)$，即：

$$Conf(X\Rightarrow Y)=\frac{Supp(X\cup Y)}{Supp(X)} \quad (2)$$

由式(1)和(2)可以看出，支持度 $Supp(X\cup Y)$体现了项目集 X 和 Y 在交易集中出现的频度，而可信度 $conf(X\Rightarrow Y)$则体现了项目集 X 和 Y 之间的关联程度。给定一个交易集 T，关联规则挖掘的任务就是从 T 中找出支持度和可信度分别大于用户给定的最小支持度(minisupp)和最小可信度(miniconf)的关联规则。

关联规则挖掘问题可以分解为前后相连的两个子问题[5,6]：

第一步：找到所有的支持度大于最小支持度(minisupp)的项集(itemset)，称为频繁项集(Frequent Itemset)。

第二步：从上述频繁项集中生成期望得到的规则。这一步相对简单。对于每一个频繁项集 L，可以产生 L 的所有子集。对每个子集 α，如果有 $Supp(L)/Supp(\alpha)\geqslant miniconf$，就可以产生一个形如 $\alpha\Rightarrow(L-\alpha)$的规则。我们可以通过考察 L 的所有子集来产生所有符合要求的规则。

我们可以看出问题的难点在第一步，上面提到的 Apriori-Tid 算法就是用于产生频繁项集的典型算法，本文将深入探讨基于 Apriori 算法的频繁项集关联规则算法。

2　基于 Apriori 算法的多循环关联规则挖掘过程

关联规则挖掘实质上就是在满足用户给定的最小支持度的频繁项集中，找出所有满足最小置信度的关联规则[5~9]。它可以分为如下几个步骤。

①预处理和发现与挖掘任务有关的资料。根据具体问题的要求对数据库进行相应的操作，从而构成规格化的数据库 DB。

②针对 DB，求出所有满足最小支持度的项集，即频繁项集。由于一般情况下我们所面临的数据库都比较大，因此此步是算法的核心。主要采用以下方式进行：

模式过滤：借助模板或其它类型的选择工具来定义待发现规则的类型。

统计过滤：删除模式过滤后在统计意义上不重要的规则。通过允许用户设定某些适当的统计参数或选择适当的技术来完成。

*)基金项目：国家自然科学基金重点科研课题，项目编号：70031010。袁军鹏　博士生，讲师，研究方向：管理信息系统，数据挖掘。朱东华　美国佐治亚理工学院技术政策与评估中心资深研究员，教授，博士导师，研究方向：科技管理，技术预测，数据挖掘，人工智能，信息系统。

维普资讯 http://www.cqvip.com

语意过滤：删除掉在语意上没有意义的规则。

③生成满足最小置信度的规则，形成规则集 RT。

④规则的评估与解释。强调用户参与，用户手工选择感兴趣的规则，形成最后的规则集。

⑤规则的用户化呈现。用数据可视化技术，以用户容易理解的方式把挖掘到的规则显示出来。

3 多循环方式典型算法分析

Agrawal 等人首先提出了挖掘顾客交易数据库中项集间的关联规则问题，其核心方法是基于频繁集理论的递推方法[5]。以后诸多的研究人员对关联规则的挖掘问题进行了研究。

此类算法包括 Agrawal 等人提出的 AIS[5]，Apriori、AprioriTid 和 AprioriHybrid[6]、Park 等人提出的 DHP[10]，Savasere 等人的 PARTTTION[11]以及 Toivonen 提出的抽样算法 Sampling[12]等等。其中最有效和有影响的算法为：Apriori，DHP 和 PARTTTION。

3.1 AIS 算法简介

在算法 AIS 中，候选频繁项集是在扫描数据库的过程中产生的。具体地说，在对数据库进行第 k 次扫描时，候选频繁项集(其中每一个元素的元素个数不一定是 k 个，可以大于 k)是由第 k－1次扫描所产生的边界集(Frontier set)通过增加当前交易中的项得到，同时计算候选频繁项集中的元素的支持数，直到某一次扫描所产生的边界集为空时停止运算。应该注意的是，第 k 次扫描所产生的边界集要大于本次扫描生成的频繁项集。本算法的缺点在于生成的候选频繁项集太大。

3.2 Apriori、AprioriTid 和 AprioriHybrid 算法分析

算法 Apriori 和 AprioriTid 利用“在给定的交易数据库 D 中，任意频繁项集的子集都是频繁项集；任意弱项集的超集都是弱项集”这一原理对交易数据库进行多遍扫描。

Apriori 的核心思想

```
L1={large 1-itemsets};
for (k=2;Lk-1≠Φ; k++)do begin
      Ck=apriori-gen(Lk-1); //生成新的候选集
      for all transactions t∈D do begin
          Ct=subset(Ck,t); //交易 t 中包含的候选集
      for all candidates c∈Ct do
      c.count++
      end
      Lk={c∈Ck| c.count≥min-sup }
end
Answer=∪kLk;
```

首先产生频繁1-项集 L_1，然后是频繁2-项集 L_2，直到有某个 r 值使得 L_r 为空，这时算法停止。这里在第 k 次循环中，过程先利用第 k－1次扫描的结果(即大 k－1项集 L_{k-1})和函数 apriori-gen[6]产生候选频繁 k 项集 C_k，C_k 中的每一个项集是对两个只有一个项不同的属于 L_{k-1} 的频集做一个(k－2)-连接来产生的。C_k 中的项集是用来产生频集的候选集，最后的频集 L_k 必须是 C_k 的一个子集。C_k 中的每个元素需在交易数据库中进行验证来决定其是否加入 L_k，这里的验证过程是算法性能的一个瓶颈。这个方法要求多次扫描可能很大的交易数据库。

其中，apriori-gen 函数的功能是从第 k－1次遍历数据库后找出频繁项集集合 L_k，产生出第 k 次遍历所要计数的长度为 k 的候选项集集合 C_k，并且要保证 Ck 中项集的所有 k－1项子集都是频繁项集。subset 函数利用 hash 函数找出包含在交易 t 中的所有候选项集。

Apriori-Tid 算法主要分为两步：(1)简单统计所有仅含一个元素的项集的出现频率，并找出那些支持度不小于最小支持度的项集，即为一阶频繁项集；(2)对数据源进行循环处理直到没有新的频繁项集生成为止。其循环过程是：第 k 步中，根据第 k－1步生成的(k－1)阶频繁项集，运用函数 apriori-gen 产生 k 阶候选集，然后从候选集中根据最小支持度选出 k 阶频繁项集。具体的算法参见文[6，13～15]。

由于充分利用剪枝技术(Pruning)[6,13]，这两个算法所产生的候选频繁项集要比算法 AIS 小得多，从而提高了算法的效率。算法 AprioriTid 还有另外一个特点，即仅在第一次扫描时用交易数据库 D 计算候选频繁项集的支持数，其它各次扫描用其上一次扫描生成的候选交易数据库 D' 计算候选频繁项集的支持数。在最后的几次扫描中，D' 大小要远远小于 D，减小了 I/O 操作时间，提高了算法的效率。

算法 AprioriHybrid 是算法 Apriori 与算法 AprioriTid 的结合，当候选交易数据库 D' 不能完全容纳于内存时用算法 Apriori，当内存能够完全容纳候选交易数据库 D'，则用算法 AprioriTid。

3.3 DHP 算法分析

通过实验可以发现寻找频集主要的计算是在生成频繁2-项集 L_k 上，Park 等利用哈希(Hashing)技术有效地改进了候选频繁项集的生成过程[10,16]，产生了比前述算法更小的候选频繁项集(对大2候选集尤为明显)，同时也缩减了交易数据库的大小，减小了 I/O 操作时间，其效率比算法 Apriori 有明显提高。该算法的缺点是：在第 k 次扫描数据库时需要建立关于(k＋1)项集的哈希表，需要额外的时间和空间，而且实现起来比较麻烦。

DHP 算法描述如下：

```
把 H2的所有桶初始化为0;//初始化 Hash 表
for all transactions t∈D do begin
    insert and count 1-items occurrences in a hash tree;
    for all 2-subsets x of t do
        H2[h2(x)]++;
end
L1={c|c.count≥min-sup,c 是 hash 树的一个叶子节点};
k=2;
Dk=D;
While ( | {x|Hk[x]≥min-sup} | ≥LARGE){ //生成 Hash 表
      gen-candidate(Lk-1,Hk,Ck);
      把 Hk+1的所有桶初始化为0;
      Dk+1=Φ;
      for all transactions t∈Dk do begin
      count-support(t,Ck,k,t̃);//t̃⊆t
      if(|t̃|>k)then do begin
         make-hasht(t̃,Hk,k,Hk+1,t̃);
         if (|t̃|>k)then Dk+1=Dk+1∪{t̃};
      end
      Lk={c∈Ck|c.count≥min-sup};
      k++;
}
```

3.4 PARTITION 分析

算法 PARTITION 分为两个部分。在第一部分中，算法首先将要在其中发现关联规则的交易数据库 D 分为 n 个互不相交的交易数据库 $D^1, D^2, \cdots, D^n$，D^i(i=1，2，…，n)的大小要求能够容纳在内存之中，然后将每一个分交易数据库 D^i 读入内存并发现其中的频繁项集 L^i，最后在第一部分结束时将所有分交易数据库的频繁项集合并成为一个在交易数据库 D 中的潜在频繁项集 $PL=\bigcup_{i=1}^{n}L^i$；算法第二部分计算潜在频繁项集 PL 在交易数据库 D 中的支持数，并得出频繁项集 L[11]。该算法只对交易数据库 D 扫描两次，大大减少了 I/O 操作，降低算法对内存的需求同时提高并行性，从而提高了算法的效率。

Partition 算法描述如下：

```
P=partition-database(D);
n=分区数;
for i=1 to n do begin //第一步
    read-in-partition(pi∈P)
```

维普资讯 http://www.cqvip.com

```
    L'=gen_large_itemsets(p_i)
end
for(i=2;L_i^j≠Φ;j=1,2,…,n;i++)do begin //合并
C_i^G=∪_{j=1,2,…,n}L_i^j
    end
for i=1 to n do begin //第二步
  read_in_partition(p_i∈P)
  for all candidate_itemsets c∈C^G 计算 gen_count(c,p_i)
end
L^G={c∈C^G|c.count≥min_sup}
answer=L^G
```

3.5 Sampling 算法分析

算法 Sampling 设计思想为：先使用从数据库中抽取出来的采样得到一些在整个数据库中可能成立的规则，然后对数据库的剩余部分验证这个结果。对交易数据库 D 进行随机抽样得到抽样交易数据库 D'，先以小于用户给定的最小支持度的支持度发现存在于 D' 中频繁项集 L'，再在剩下的数据库 D－D' 中继续计算 L' 中各元素的支持数，最后以用户给定的最小支持度计算出频繁项集 L。在大部分情况下，可以用此方法发现正确的频繁项集，但有时可能会漏掉一些频繁项集，这时可以对 D 进行第二次扫描以得出漏掉的频繁项集[12]。此算法在大部分情况下可以只对 D 进行一次扫描就得出频繁项集，最坏的情况对 D 进行两次扫描，因而算法效率较高。

算法 Sampling 描述如下：

```
//draw a sample and find the local large itemsets in the sample
1)Ds=a random sample drawn from D;
2)PL=Apriori(I,Ds,s);
//first scan counts the candidates generated from PL
3)C=PL∪BD(PL);
4)count(C,D);
//second scan counts additional candidates if there are misses in BD
  (PL)
5)ML={x|x∈BD(PL),x.count≥s×|D|};//ML are the misses
6)if ML≠∅then //MC are new candidates generated from the misses
7)MC={x|x∈C,x.count≥s×|D|};
8)repeat
9)MC=MC∪BD(MC);
10)until MC doesn't grow;
11)MC=MC-C;//itemsets in C have already been counted in scan
  one
12)count(MC,D);
13)return L={x|x∈C∪MC,x.count≥s×|D|};
```

4 对 Apriori 的改进

上面介绍的都是基于 Apriori 的频集方法，即使进行了优化，但是 Apriori 方法的一些固有的缺陷还是无法克服：

(1)可能产生大量的候选集。当长度为1的频集有10000个的时候，长度为2的候选集个数将会超过1000万。还有就是如果要生成一个很长的规则的时候，要产生的中间元素也是巨大量的。

(2)无法对稀有信息进行分析。由于频集使用了参数 minisupp，因此就无法对小于 minisupp 的事件进行分析；而如果将 minisupp 设成一个很低的值，那么算法的效率就成了一个很难处理的问题。

(3)会产生大量冗余的、虚假的和非采掘者关心的关联规则。基于 Apriori 的频集方法衡量和生成关联规则的标准主要有两个，即支持度和置信度。但仅用这两个标准来衡量关联规则，往往会发现大量冗余的、虚假的和非采掘者关心的关联规则。

(4)忽略了各属性之间的权重。基于 Apriori 的频集方法暗含了这样一个事实：数据库中各属性之间都是平等的，无重要、次要之分，然后在此基础上再进行规则的挖掘。但实际上，如果从某种角度出发(比如利润的大小)，数据库中各属性的重要程度是不一样的。

(5)忽略了用户的参与，只能发现所有的关联规则。

下面将介绍几种方法，分别用于解决以上问题。

4.1 FP-growth 的方法解决候选集过大问题

在文[17]中提到了解决问题1的一种方法——FP-growth 方法。该方法使用了分而治之的策略：在经过了第一次扫描之后，把数据库中的频集压缩进一棵频繁模式树(FP-tree)，同时依然保留其中的关联信息。随后再将 FP-tree 分化成一些条件库，每个库和一个长度为1的频集相关。然后再对这些条件库分别进行挖掘。当原始数据量很大的时候，也可以结合划分的方法，使得一个 FP-tree 可以放入主存中。实验表明，FP－growth 对不同长度的规则都有很好的适应性，同时在效率上较之 Apriori 算法有巨大的提高。具体算法请参见文[17]。

4.2 对稀有信息的分析

文[18]提出了解决第二个问题的一种思路：Apriori 算法得出的关系都是频繁出现的，但是在实际的应用中，可能需要寻找一些高度相关的元素。在 Apriori 算法中，起决定作用的是支持度，而我们现在将把可信度放在第一位，挖掘一些具有非常高可信度的规则。整个算法基本上分成3个步骤：计算特征、生成候选集、过滤候选集。在3个步骤中，关键的地方就是在计算特征时 Hash 方法的使用。基本的方法有两类：Min_Hashing(MH)和 Locality_Sensitive_Hashing(LSH)。最后的实验数据也说明这种方法的确能产生一些有用的规则。具体算法请参见文[18]。

4.3 加入兴趣度等解决冗余问题

文[19]给出的一个例子来说明仅仅利用用户定的最小支持度和最小置信度来生成关联规则，则往往会发现大量冗余的、虚假的和非采掘者关心的关联规则。如考察某个学校5，000名学生的早晨活动情况。调查结果表明：60%学生打篮球，75%的学生吃麦片粥，40%的学生既打篮球又吃麦片粥。若我们将最小支持度和最小置信度分别定为40%和60%，则我们可以发现关联规则：打篮球⇒吃麦片粥，其支持度和置信度分别为40%和66.6%。该规则实际上是一个虚假的规则，因为吃麦片粥的学生比例为75%，既高于最小支持度60%，又高于规则的支持度66.6%。因而该规则提供的信息没有任何价值。

这种不足已经引起了许多学者的注意，他们提出了一些采掘有趣和实用的关联规则的算法[20~25]，但还没有形成一个统一的标准。我们认为有必要将兴趣度这个标准加入到关联规则的定义之中，这样才符合数据采掘的宗旨和减少关联规则采掘的盲目性。

文[26]也提出了一种通用的由最大频繁项目集生成无冗余关联规则的 GNRR(GenerateNon-RedundantRules)算法，利用规则之间的冗余关系，按一定顺序挖掘不同的规则，消除了规则之间的冗余性，使发现的规则数目呈指数倍减少。

4.4 加权关联规则

文[27~33]分别提出了对属性、记录、项目和时间序列等进行加权挖掘的方法。下面简述如下，具体算法请参见相应文献。

文[30]提出布尔型加权关联规则和广义模糊型加权关联规则挖掘算法，这里通过对属性加权，可以使用户发现那些以前被忽视但可能又很重要的规则，同时由于引进了模糊的概念，使得最终得到的规则更容易被人理解。

文[31]提出了“记录加权型关联规则挖掘”算法。文献指出“在所有的历史记录中，不同的记录对预测未来模式的参考价值是不同的”。所以，在关联规则挖掘过程中，不同的记录对最终规则的生成应该有不同的贡献度，即不同的记录应有不同的权重。可以根据历史记录中某一个或几个属性的情况来

维普资讯 http://www.cqvip.com

为每一条记录计算出它的权重，然后结合记录的权重使用改进后的关联规则挖掘算法以获得需要的关联规则。

文[32]根据现实世界数据库中不同的项目往往有着不同重要性的实际情况，对关联规则发现问题进行了推广，提出了加权关联规则发现问题。为反映各个项目的不同的重要性，对项目引入了权值。由于项目权值的引入，频繁项目集的子集不再一定是频繁的。为此，文献提出了项目的 k-支持期望概念，并由此提出了加权关联规则的发现算法 DWAR。实验结果表明，DWAR 算法性能良好，具有较好的可扩放性。

文[33]指出实际应用中，用户更关心近期发生的数据，即历史越久远的数据对规则的影响应该越小，应当削弱这些数据对规则的影响，为此，提出了垂直加权关联规则的问题；另外，用户有时可能希望加强或削弱某些项目对规则的影响，即所谓的水平加权关联规则。最后，提出了混合加权关联规则的问题，并给出了一个解决该问题的算法 MWAL，实验证明了 MWAL 算法的有效性。

4.5 人机交互，提高用户参与程度

文[29]提出了用布尔表达式表示的有项目限制的关联规则挖掘方法，但仍忽略了用户的实时交互性。

文[34]指出，将关联规则挖掘集中于数据中某些特定的部分的技术应用显得非常必要，领域知识在数据库中的应用成了必然。领域知识就是在数据库中未明确表达，而在某领域内达成共识的信息知识。在数据准备与预处理阶段，可以利用领域知识排除与关联规则发现无关的数据库中的记录或数据项，有效地降低问题的维数，缩减数据库的大小，使设计出的采掘算法更加有效。在数据挖掘的产生假设阶段，用领域知识排除假设中不必要的条件以优化假设，减少从数据库中发现有用信息的搜索时间。在发现新模式阶段，利用领域知识及对规则的关注程度对无意义的候选关联规则进行修剪，以提高挖掘效率。但领域知识应该认真使用以防阻碍一些有用规则的发现。

文[35]指出用户趋向于对某一类或几类项目感兴趣，提出基于频繁集的交互式关联规则挖掘，提供了友好、方便的界面，用户可从当前的频繁集按实际需要选择某些(或某一)感兴趣的项目类(或子类)，并给出所期望的最小支持度值和最小可信度值，以求能发现某些新奇的、反常的关联规则。在满足用户交互功能的同时，交互式免去了用户不感兴趣的项目间关联规则的挖掘，因而极大地提高了挖掘速度。

结束语 本文对近年来关联规则采掘中循环式挖掘进行了回顾，并对其典型算法进行了分析和讨论，指出了基于 Apriori 算法的关联规则挖掘的不足，并归纳和总结了国内外学者对该算法的改进。

对于关联规则的发展，我们觉得可以在下面一些方向继续深入：研究在处理极大量的数据时，如何提高算法效率的问题；在挖掘的过程中，提供一种与用户进行交互的方法，将用户的领域知识结合在其中；生成结果的可视化问题。

参 考 文 献

1 Benoit G. Data mining[J]. Annual Review of information Science and Technology, 2001. 265～310
2 Dunham M H, et al. A Survey of Association Rules
3 Ou-Yang, Wei-min, Cai, Qing-sheng. Researches on discovery of association rules[J]. Computer Science, 1999, 3: 41～44 (in Chinese)
4 Chen M-S, Han J W, Yu P. Datamining: an overview from database perpective[C]. Shanghai: Fudan Inter-national Data Mining Conference, 1998, 11: 12～15
5 Agrawal R, Imeielinski T, Swami A. Mining association rules between sets of items in large databases[C]. In: Proc. of ACM SIGMOD Conf. on Management of Data, Washington D. C, May 1993. 207～216
6 Agrawal R, Srikant S. Fast Algorithms for Mining Association Rules in Large Databases. In: Proc. of the 20th Intl. Conf. on Very Large Data Bases, Sep. 1994. 487～499
7 Srikant R R, Agrawal R. Mining quantitative association rules in large relational tables. ACM SIGMOD Record, 1996, 25(2): 1～12
8 Ng R T, Han J. Efficient and Effective Clustering Methods for Spatial Data Mining. In: Proc. of the 18th Intl. Conf. on Very Large Data Bases, Sep. 1994. 144～155
9 Park J S, Yu P S, Chen M-S. Mining association rules with adjustable accuracy. In: Proc. of the sixth Intl. conf. on Information and knowledge management, Las Vegas, Nevada, United States, 1997. 151～160
10 Park J S, Chen M S, Yu P S. An effective hash-based algorithm for mining association rules. In: Proc. 1995 ACM SIGMOD Intl. Conf. Management of Data, SanJose, CA, 1995. 175～186
11 Savasere A, Omiecinski E, Navathe S. An efficient algorithm for mining association rules. In: Proc. of the 21st Intl. conf. on very large databases, Zurich, Switzerland, Sept. 1995. 432～444
12 Toivonen H. Sampling large databases for association rules. In: Proc. of the 22nd Intl. conf. on Very Large Databases. Mumbai, India, 1996. 134～145
13 Agrawal R, Srikant R R. Fast algorithms for mining association rules. Business Intelligence, 1998. 560～564
14 Sunita S, Shiby T, Rakesh A. Integrating Mining with Relational Database Systems: Alternatives and Implications. In: Proc. ACM SIGMOD Intl. Conf. on Management of Data, SIGMOD 1998, Seattle, Washington, USA
15 Srikant R, Agrawal R. Mining Quantitative Association Rules in Large Relational Tables. In: Proc. of the 1996 ACM SIGMOD Intl. Conf. on Management of Data, Montreal, Quebec, Canada, 1996. 1～12
16 Park J S, Chen M-S, Yu P S. Efficient Parallel Data Mining for Association Rules. In: Proc. of the Intl. Conf. on Information and Knowledge Management, Baltimore, Maryland, 1995. 31～36
17 Han J, Pci J, Yin Y. Mining Frequent Patterns without Candidate Generation. In: Proc. of the 2000 ACM SIGMOD Intl. Conf. on Management of Data, Dalas, TX, 2000
18 Cohen F, Datar F, Fujiwara S. Finding Interesting Associations without Support Pruing
19 Chen M S, et al. Data mining: an overview from database perspective. IEEE Transactions on knowledge and data engineering, 1996, 8(6): 866～883
20 Srikant R, Agrawal R. Mining Generalized Association Rules. In: Proc. of the 21nd Intl. Conf. on Very Large Databases, Zurich, Swizerland, 1995. 407～419
21 Brin S, Motwani R, Silverstein C. Beyond Market Baskets: Generalizing Association Rules to Correlations. In: Proc. of the ACM SIGMODConf. 1997. 265～276
22 Aggarwal C C, Sun Z, Yu P S. Online Algorithms for Finding Profile Association Rules. In: Proc. of the ACM CIKM Conf. 1998. 86～95
23 Aggarwal C C, Wolf J L, Yu P S, Epelman M. Online Generation of Profile Association Rules. In: Proc. of the Intl. conf. on Knowledge Discovery and Data Mining, Aug. 1998
24 Brin S, Motwani R, Ullman D, Tsur S. Dynamic Itemset Counting and Implication Rules for Market Basket Data. In: Proc. of the ACM SIGMOD Conf. 1997. 255～264
25 周欣，沙朝锋，等. 兴趣度——关联规则的又一阈值. 计算机研究与发展，2000(5)
26 高峰，谢剑英. 一种无冗余的关联规则发现算法. 上海交通大学学报，2001(2)
27 Bayardo R J, Agrawal R, Gunopulos D. Constraint-based rule mining in large, dense databases. In: Proc. of the 15th Int l Conf. on Data Engineering. 1999. 188～197
28 Cai C H, et al. Mining association rules with weighted items. In: Proc. Of the Int l Database Engineering and Applications Symposium. 1998. 768～774
29 Srikant R, Vu Q, Agrawal R. Mining association rules with item constraints. In: Proc. of the Third Int l Conf. in Knowledge Discovery in Databases and DataMining. 1997. 67～73
30 陆建江. 数据库中布尔型及广义模糊型加权关联规则的挖掘. 系统工程理论与实践，2000(2): 28～32
31 江卓军，谢康林，张文杰. 一种新的关联规则挖掘思想. 计算机工程，2002(4): 88～90
32 欧阳为民，郑诚，蔡庆生. 数据库中加权关联规则的发现. 软件学报，Vol. 12, No. 4: 612～619
33 路松峰，胡和平. 加权关联规则的开采. 小型微型计算机系统，2001(3): 347～350
34 郑泽芝. 采掘关联规则的算法及优化处理概述. 太原师范专科学校学报，2000(4): 4～6
35 赵奕，施鹏飞，熊范纶. 基于频繁集的多层次交互式关联规则挖掘. 上海交通大学学报，2000(5): 695～698