维普资讯 http://www.cqvip.com

关系数据库 关联规则 数据采掘 数据挖掘 数据库

计算机科学 1999 Vol. 26 №. 1

62-64

自上而下的关联规则采掘方法

Mining Association Rules Using Top-down Method

何 嵩 田振军

(复旦大学计算机科学系 上海 200433)

TP311.13

Abstract Mining association rules from a data-set is an important branch in data mining area. In a bottom-up method, we have to adjust the thresholds at different levels many times for good effect, and the process of adjustment is hard to control. On the other hand, the bottom-up method can only assign field-names to the values at the same level. In this paper, we first raised some definitions about the specialization and also some theorems. Then we gave our top-down method of the association rule mining on this base. This method can effectively resolve these two problems with the same good efficiency.

Keywords Data Mining, Knowledge Discovery in Databases, Association Rule, Specialization

1. 前言

在数据集中采掘关联规则是数据采掘的一个重要内容.[2]中提出了一种自底向上的数据采掘方法,即把数据集的各个属性从概念层次的最底层向高层进行"泛化",但这种方法有二个缺点:首先,它的每一层都有一个阈值,但这些阈值之间是密切相关的;并且,对于不同数据集,这些阈值之间的关系也不同。在数据采掘的过程中,要调整这些阈值以取得满意的效果需要很多次尝试、分析和重运行。第二,每个字段的赋值有着必然的联系;即,若一个字段赋值于它所对应的概念层次上的倒数第二层,则其它的任何一个字段都被赋值于倒数第二层的常量。

本文提出了一个自上而下采掘关联规则的策略,较好地克服了这两个缺点,并具有同样满意的效率。

2. 问题背景

在关系数据库中采掘关联规则一般先要选择好数据集。比如说,若在采掘学生档案库中的专业、课程及成绩之间的关联规则,则可以选取一个关系 R (Sno, Speciality, Course, Grade),这个关系可能是对数据库中的若干个表作联结和选择后得到的。

在上述数据集中,我们期望采掘得到诸如"软件专业学生的计算机类课程有 37%是优秀的"之类的知识。这一知识是用如下的关联规则来描述的:

$$Speciality = 'Software' \wedge Course = 'CS_Course' \rightarrow Grade = 'Excellent' (37\%) \quad (2.1)$$

对于数据集 R 上的一条关联规则 B→H,其中 B 和 H 都是'字段名=某个值'的合取式,则它的可信度定义为:

$$confidence = |\sigma_{B \wedge H}(R)| / |\sigma_B(R)|$$

其支持度定义为:

$$support = |\sigma_{B \wedge H}(R)| / |R|$$

规则(2.1)中的 37%是其可信度,其实它表示规则的正确程度.

对于支持度和可信度太小的规则,用户一般是不感兴趣的,因此可以提供两个阈值,在采掘过程中我们只关心支持度和可信度大于(或等于)阈值的关联规则。

在采掘过程中,有时用户并不关心太具体的详细内容,而更在意一些更高层次的统计信息。例如:用户可能更关心计算机类课程的整体情况,而不关心《数据结构》课程的情况,这样,就需要把不同计算机类课程作为同一门课程来处理。我们可以用概念层次来描述某个具体常量在不同层次上的抽象常量值。

给出一个数据集,及其每个字段的概念层次、可信度和支持度的阈值,我们要采掘出支持度和可信度都大于(或等于)相应阈值的关联规则。

维普资讯 http://www.cqvip.com

3. 若干定义和定理

特化：设数据集中有 n 个字段，则该数据集的一个特化是指一个 n 元组$(a_1,a_2,\cdots,a_n)$，它表示把数据集中的第 i 个字段赋为 a_i，其中 a_i 是其概念层次的某个值。若不对第 i 个字段赋值，则 $a_i=$ 'any'。例如：对于数据集 R（Sno，Speciality，Course，Grade），特化（'any'，'Software'，'CS_Course'，'Excellent'）表示合取式：Speciality='Software' ∧ Course='CS_Course' ∧ Grade='Excellent'。

有效特化：设特化$(a_1,a_2,\cdots,a_n)$的合取式为 e，支持度阈值为 s。若$\frac{|\sigma_e(R)|}{|R|}\geqslant s$，则称$(a_1,a_2,\cdots,a_n)$为有效特化。

特化的包含关系：设有特化$(a_1,a_2,\cdots,a_n)$和$(b_1,b_2,\cdots,b_n)$，若对于所有的 $1\leqslant i\leqslant n$，a_i 所表示的概念包含了 b_i，则称$(a_1,a_2,\cdots,a_n)$包含$(b_1,b_2,\cdots,b_n)$，记作$(a_1,a_2,\cdots,a_n)\supset(b_1,b_2,\cdots,b_n)$。

有效特化定理：设有有效特化$(b_1,b_2,\cdots,b_n)$，$(a_1,a_2,\cdots,a_n)\supset(b_1,b_2,\cdots,b_n)$。则$(a_1,a_2,\cdots,a_n)$也是有效特化。

证明：∵$(b_1,b_2,\cdots,b_n)$为有效特化。

∴由定义，若$(b_1,b_2,\cdots,b_n)$的合取式为 e，支持度阈值为 s，则$\frac{|\sigma_e(R)|}{|R|}\geqslant s$。

∵$(a_1,a_2,\cdots,a_n)\supset(b_1,b_2,\cdots,b_n)$

∴若$(a_1,a_2,\cdots,a_n)$的合取式为 e'，则显然$|\sigma_{e'}(R)|\geqslant|\sigma_e(R)|$。

∴$|\sigma_{e'}(R)|/|R|\geqslant|\sigma_e(R)|/|R|\geqslant s$，得证。

特化项数：设有特化$(a_1,a_2,\cdots,a_n)$，其中 $a_i\neq$ 'any' 的项数称为特化项数。

特化的直接包含：设有两个特化$(a_1,a_2,\cdots,a_n)$和$(b_1,b_2,\cdots,b_n)$，若存在某个 k，使得 $a_i=b_i(i\neq k)$，而在概念层次上 a_k 是 b_k 的儿子，则称$(b_1,b_2,\cdots,b_n)$直接包含$(a_1,a_2,\cdots,a_n)$。

有效特化的项数定理：设有有效特化$(a_1,a_2,\cdots,a_n)$，直接包含它的有效特化的数目等于$(a_1,a_2,\cdots,a_n)$的特化项数。

证明：很显然，直接包含$(a_1,a_2,\cdots,a_n)$的特化的数目等于$(a_1,a_2,\cdots,a_n)$的特化项数。而所有直接包含$(a_1,a_2,\cdots,a_n)$的特化都是有效特化，这可由有效特化集定理证得。

特化和：设有效特化$(a_1,a_2,\cdots,a_n)$，设 a_i 出现在概念层次的第 m_i 层（根为第 0 层），称 $\sum_{i=1}^{n} m_i$ 为$(a_1,a_2,\cdots,a_n)$的特化和。

和相同的有效特化的集合：把所有特化和为 k 的有效特化组成集合记为 $S(k)$。我们将在下一节中递归地构造 $S(k)$。构造结束后就求出了数据集的所有有效特化。

4. S(k)的递归构造

在下面的算法中，每个特化都维护了一个计数器。

```
S(0)={('any','any',…,'any')};
i=0;
while S(i)≠φ do
begin
  i=i+1;
  S(i)=φ;
  For all(a1,a2,…,an)∈S(i-1) do  ……… (*)
    For j=1 to n do
      For aj 的所有儿子 b do
      If ((a1,a2,…aj-1,b,aj+1,…,an)∈S(i) then
          (a1,a2,…,aj-1,b,aj+1,…,an)的计数器+
          1
      else
          把(a1,a2,…,aj-1,b,aj+1,…,an)加入 S(i)
          且把其计数器设为 1;
  For all (a1,a2,…,an)∈S(i) do
    If (a1,a2,…,an)的计数器小于其特化项数
    then
      Delete (a1,a2,…,an) from S(i)----(**)
    else if(a1,a2,…,an)的支持度<阈值 then
      Delete(a1,a2,…,an) from S(i)----(***)
End
```

定理：上述算法正确地计算了 $S(i)$。

证明：可用归纳法证明。证明每次构造的 $S(k)$ 就是和为 k 的有效特化的集合。

$k=0$，显然成立。

现证明 $k>0$ 是的情况：假设 $S(k-1)$ 就是所有特化和为 $k-1$ 的有效特化的集合。算法先算出 $S(k-1)$ 所有的直接包含的特化，设之为 $S'(k)$，而所有算出的这些特化的特化和都为 k。若存在不属于 $S'(k)$ 且特化和为 k 的特化，则直接包含它的特化必然不属于 $S(k-1)$，由有效特化定理知它们必不是有效特化。所以，(*)得出了一个包含所有特化和为 k 的有效特化的集合 $S'(k)$。算法的(**)(***)为进一步删除非有效特化。(***)是很容易明白的，即将支持度小于阈值的特化删除，以保证所剩的特化都是有效特化。而(**)事实上是算法的优化。在计算 $S'(k)$ 时，每个特化都赋予一个计数值，用于标志该特化共被派生的次数，该计数值反映了 $S(k-1)$ 中能派生这个特化的所有有效特化的数目，由有效特化的项数定理，若该特化的计数值小于特化项数 k，则该特化不是有效特化。这样，就在用阈值

维普资讯 http://www.cqvip.com

筛选有效特化之前先用这种方法删除了一部分非有效特化。减少了(* * *)扫描所有数据的开销。这样，命题就得到证明了。

5. 关联规则的生成

完成了数据集的有效特化的计算后，我们可以方便地从这些有效特化生成关联规则。

设有效特化$(a_1,a_2,\cdots,a_n)$和$(b_1,b_2,\cdots,b_n)$满足：

①若对于所有的$a_i\neq$ 'any'，有$a_i=b_i$。

②对于所有的$b_i\neq$ 'any'，要么$b_i=a_i$，要么$a_i=$ 'any'。

③$\frac{(b_1,b_2,\cdots,b_n)\text{的支持度}}{(a_1,a_2,\cdots,a_n)\text{的支持度}}=C\geqslant$可信度阈值。

则生成关联规则：

$(a_1,a_2,\cdots,a_n)$的合取式→$(b_1,b_2,\cdots,b_n)$的合取式－$(a_1,a_2,\cdots,a_n)$的合取式其可信度为C，支持度为$(b_1,b_2,\cdots,b_n)$的支持度。

例如，设二个有效特化：('any'，'any'，'any'，'Software'，'CS_Course'，'any')和('any'，'any'，'any'，'Software'，'CS_Course'，'Excellent')，则可以生成关联规则：Speciality='Software' ∧ Course='CS_Course'→Grade='Excellent'。

定理：上面的算法是正确且完备的。

证明：正确性，即用上述算法得出的规则的支持度和可信度都大于(或等于)阈值。由定义，规则"$(a_1,a_2,\cdots,a_n)$的合取式→$(b_1,b_2,\cdots,b_n)$的合取式—$(a_1,a_2,\cdots,a_n)$的合取式"的支持度即为$(b_1,b_2,\cdots,b_n)$的支持度，显然超过其阈值。它的可信度为$(b_1,b_2,\cdots,b_n)$的支持度/$(a_1,a_2,\cdots,a_n)$的支持度。由③可知符合要求。所以，正确性得证。

完备性，即所有关联规则均可由算法得到，对于任何一条满足要求的关联规则$A\rightarrow B$，其支持度为S，可信度为C，它们都超过阈值。

设$A+B$所对应的特化为$(b_1,b_2,\cdots,b_n)$，根据定义，其支持度也为S，即超过阈值。再设A所对应的特化为$(a_1,a_2,\cdots,a_n)$，显然有$(a_1,a_2,\cdots,a_n)$包含$(b_1,b_2,\cdots,b_n)$，所以其支持度$S'\geqslant S$，也超过阈值。也即$(a_1,a_2,\cdots,a_n)$和$(b_1,b_2,\cdots,b_n)$都是有效特化。

显而易见，$(a_1,a_2,\cdots,a_n)$和$(b_1,b_2,\cdots,b_n)$满足①②，规则的可信度$C=S/S'\geqslant$阈值，从而$(a_1,a_2,\cdots,a_n)$和$(b_1,b_2,\cdots,b_n)$满足③。也就是说，这条规则一定由该算法得到。完备性得证。

6. 算法效率分析

一般来说，数据采掘的工作是在大规模数据库上进行的。相对于大规模的数据库来说，概念层次的层数和数目是很小的，我们可以假设它们是常量。设概念层次的层数为L，在采掘中使用了N个概念层次。一般来说，N不大于数据集的字段数。因为用户可能没有为某个字段给出概念层次，那么第4节中的算法中*while*语句的循环次数最多为LN次。在*while*语句体中，我们做了如下工作：

1. 构造$S(t)$

2. 删减$S(t)$中的非有效特化(利用有效特化项数定理)

3. 计算$S(t)$中各个特化的支持度，并进一步删减$S(t)$中的非有效特化

1、2以及第5节中的生成关联规则的过程都是在内存中完成的；而3需要对数据集做一次扫描完成，它要访问外存，并可能要从数据库服务器中传输数据。因此，本文中的方法主要工作量在于对数据集的扫描。我们一般用扫描数据集的次数作为评价算法复杂度的标准[2]。这样，该方法的复杂度为LNS，即$O(S)$，其中S为数据集的大小。它与[2]中自底向上的算法具有同样、甚至更高的效率。

参 考 文 献

1 FU Y, Han J. Meta-Rule-Guided Mining of Association Rules in relational databases. in: Proc. 1995 Intl Workshop on Knowledge Discovery and Deductive and Object-Oriented Databases (KDOOD'95). Singapore, Dec. 1995

2 Han J, Fu Y. Discovery of multiple-level association rules from large databases. In: Proc 1995 Int. Conf. Very Large Data Bases. Zurich, Switherland. Sept. 1995