基于遗传算法的长频繁项集挖掘方法

王 伟[1],高 亮[1],吴 涛[1,2]
(1.安徽大学 数学与计算科学学院,安徽 合肥 230039;
2.安徽大学 智能计算与信号处理教育部重点实验室,安徽 合肥 230039)

摘 要:在数据挖掘中,关联规则的挖掘是一项核心内容,且规则的生成主要集中在如何寻找频繁项集上。一般情况下,寻找频繁项集是困难的,且项数越多越难,而最长频繁项集隐含了几乎所有的频繁集,因此,寻找频繁项集就转化为寻找最长频繁项集。文中利用遗传算法的全局最优和海量数据处理性来搜索最长频繁项集,该法只需扫描一次数据库,大大节约了时间。实验也说明了该算法的可行性。

关键词:数据挖掘;关联规则;频繁项集;遗传算法
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2008)04-0019-03

A Method of Mining Long Frequent Itemset Based on Genetic Algorithm

WANG Wei[1], GAO Liang[1], WU Tao[1,2]
(1. School of Mathematics and Computational Science, Anhui University, Hefei 230039, China;
2. Ministry of Education Key Lab. of Intelligent Computing & Signal Processing, Anhui Univ., Hefei 230039, China)

Abstract: Mining the association rules is the key content in data mining, and the rules generated mainly focused on how to find frequent item sets. In the ordinary circumstances, it is difficult to find frequent item sets, especially, the more of item set the more difficult. The longest frequent item sets contains almost all of the frequent item sets, so, how to find frequent item sets can transform to find the longest frequent item sets. Use genetic algorithms' global optimum and massive data search to mine the longest frequent itemset. This method scans the database only once, so it can greatly save the time. Finally, experiment shows the feasibility of the method.

Key words: data mining; association rules; frequent itemsets; genetic algorithm

0 引 言

关联规则挖掘是数据挖掘的重要内容。Apriori算法是其中最有影响的一个算法,其主要可分为两步:

(1)由支持度,产生频繁集;

(2)基于置信度,产生强关联规则[4]。

第(1)步产生频繁集要多次扫描数据库,且要产生大量的候选集。如有10^4个频繁1—项集,则要产生10^7个候选2—项集,并对候选集——计数[1],这直接影响了算法运行效率。而最长频繁项集已隐含了基本所有频繁集,而且许多应用也只需要挖掘最长频繁集,因此挖掘最长频繁项集对数据挖掘有重要意义。

目前,对Apriori算法的优化措施已有多种,比如基于散列技术、事务压缩、划分等来提高Apriori算法的运行效率,但这都未能对算法有本质的提高。文中结合遗传算法对海量数据搜索的长处和全局最优性来寻找最长频繁项集。其基本思想是:先将数据库映射为一个矩阵,对数据库的扫描也就对应矩阵运算,这样可以避免多次扫描数据库,提高寻找频繁项集的效率;然后用遗传算法对矩阵进行优化,从中找到最长频繁项集。

1 基本概念

设$I=\{I_1,I_2,\cdots,I_m\}$是事务数据库D中的项,也即项集。D中的每个事务为T,且每一事务都有一个标识符,记为TID。设一项集A,则事务T包含A,当且仅当$A\subseteq T$。

收稿日期:2007-07-21
基金项目:国家自然科学基金资助项目(60475017,60675031);安徽省自然科学基金资助项目(050420208);安徽省高等学校省级自然科学研究项目(2006KJ244B);安徽大学学术创新团队和安徽大学人才队伍建设经费资助项目

作者简介:王 伟(1984-),男,河南信阳人,硕士研究生,研究方向为智能计算与信息处理;吴 涛,博士,副教授,硕士生导师,主要从事机器学习、智能计算及其应用的研究。

关联规则是形如 $A \Rightarrow B$ 的蕴含式，其中 $A \subset I$，$B \subset I$，并且 $A \cap B = \varnothing$。规则 $A \Rightarrow B$ 具有支持度 s，s 是数据库 D 中事务包含 $A \cup B$ 的百分比。

包含 k 个项的项集称为 k— 项集。满足最小支持度 min_ sup 的 k— 项集称为频繁 k— 项集，记为 L_k。

Apriori 算法使用一种称作逐层搜索的迭代方法，k— 项集用于搜索 $(k+1)$— 项集，且通过 Apriori 的主要性质来压缩搜索空间：

性质 1：频繁项集的所有非空子集也必须是频繁的，或非频繁项集的超集一定是非频繁的，这就是 Apriori 性质的反单调性。

性质 2：任一频繁项集 F，则在事务数据库 D 中，一定存在一事务 T，使得 $F \subseteq T$。

基于以上两个性质，Apriori 算法分别通过连接步和剪枝步来搜索频繁 k— 项集 L_k，其具体步骤可描述如下：

(1) 连接步：为了找 L_k，通过 L_{k-1} 与自己连接产生候选 k— 项集的集合。设 l_1 和 l_2 是 L_{k-1} 中的项集，记号 $l_i[j]$ 表示 l_i 的第 j 项。执行连接 $L_{k-1} \triangleright\triangleleft L_{k-1}$，其中 L_{k-1} 的元素是可连接的，若它们前 $(k-2)$ 个项相同。即是，L_{k-1} 的元素 l_1 和 l_2 是可连接的，如果 l_1 和 l_2 的前 $(k-2)$ 个项相同，最后一项不同，则连接 l_1 和 l_2 产生的结果项集是 $l_1[1]l_1[2]\cdots l_1[k-1]l_2[k-1]$。

(2) 剪枝步：设 C_k 是 L_k 的超集，即所有的频繁 k— 项集都包含在 C_k 中，对 C_k 中的候选集计数来确定 L_k，然而 C_k 可能很大，为了压缩 C_k，利用性质 1，对 C_k 进行剪枝，从 C_k 中删除那些不频繁的项集，并且该项集的所有超级也都可删除。

根据以上两步，首先找出频繁 1— 项集的集合，该集合记为 L_1，L_1 用于找频繁 2— 项集的集合 L_2，而 L_2 用于找 L_3，如此下去，直到不能找到频繁 k— 项集。这样找每个 L_k 就需要扫描一次数据库，而且产生的候选集呈指数增加，还要对其进行一一计数和筛选，影响了运行效率，这也是 Apriori 算法的一大瓶颈[3]。

2 数据库的矩阵表示

对数据库 D，给定映射：

$$f: D \rightarrow \boldsymbol{M}$$

即 $f(D) = \boldsymbol{M} = (r_{ij})_{n \times m}$

其中 $r_{ij} = \begin{cases} 1 & I_j \in T_i \\ 0 & I_j \notin T_i \end{cases}$ $i = 1,2,\cdots,n; j = 1,2,\cdots,m$

其中 T_i 为数据库 D 中的事务，n 为事务的个数，m 为项的个数[2]。

这样在 f 的作用下，数据库 D 就被映射为矩阵 $\boldsymbol{M}$。所有对 D 的扫描均可在 $\boldsymbol{M}$ 上运算完成。比如表 1，在 f 的作用下生成矩阵 $\boldsymbol{M}$。

表 1 数据库转为矩阵示例

TID	ITEM	$\boldsymbol{M}$
001	I1,I2,I5	1 1 0 0 1
002	I2,I4	0 1 0 1 0
003	I2,I3	0 1 1 0 0
004	I1,I2,I4	1 1 0 1 0
005	I1,I3	1 0 1 0 0
006	I2,I3	0 1 1 0 0
007	I1,I3	1 0 1 0 0
008	I1,I2,I3,I5	1 1 1 0 1
009	I1,I2,I3	1 1 1 0 0

3 遗传算法及其操作

遗传算法(Genetic Algorithm，简称 GA)是通过模拟生物进化过程来完成优化搜索的，遗传算法本质上是一类建立在模拟生物进化过程基础上的随机搜索方法，其基本思想可概括为：将待优化问题的目标函数转换到某生物种群对环境的适应性，将优化变量对应生物种群的个体，将所发展的求解优化问题的算法与生物种群的进化过程类比[6]。

遗传算法与传统优化算法相比有以下特点：高鲁棒性，全局搜索能力，内在并行性。

3.1 遗传算子

(1)编码规则：文中用二进制(bit string)编码，即对一事务 T，若 $I_i \in T$，则令码串的第 i 位为 1，否则为 0。比如数据库共有 5 项，则 01001 就表示事务 $\{I_2, I_5\}$，因此对其计数就可直接利用矩阵乘法得到。比如对码串 X，$\boldsymbol{M} * X$ 为一列向量，在其中找出与 dot(x,x)(即内积) 相等元素的个数，即为事务 X 的计数。这直接利用矩阵乘法，不用一遍遍地扫描数据库，从而可大大提高计算效率。

(2)适应值函数：由于要找出数据库中满足最小支持度 min_ sup 的最长模式集，即目标函数为 max(sum (x))，x 为二进制码串，且其限制条件为 support(x) >= min_ sup。由于遗传算法默认的目标函数为最小规划，从而适应值函数(fitness) 可定义如下：

$$\text{Fitness}(x) = \begin{cases} -\text{sum}(x) & \text{support}(x) \geqslant \text{min\_ sup} \\ \text{size}(\boldsymbol{M},2) & \text{support}(x) < \text{min\_ sup} \end{cases}$$

其中 size$(\boldsymbol{M},2)$ 为矩阵的列数，也就是数据库 D 中的项数。sum(x) 对应着项集的长度。

(3)交叉算子：交叉算子主要是进行基因块之间的互换操作，通过交换两父代个体的部分信息构成后代个体，使得后代继承父代的有效模式，从而有助于产生优良个体，为了增加种群的多样性，设交叉方式为多点

交叉,即在基因的不同位置与其他基因进行交换。交叉概率设为0.85。

(4)变异算子:变异操作是通过随机改变个体中某些基因而产生新的个体,变异操作是产生全局最优的一个重要原因,有助于增加种群的多样性,通常一个较低的变异率就足以防止整个种群中任一位置的基因一直保持不变,但是概率太小则不会产生新的个体,概念太大则使GA成为随机搜索。为了防止遗传算法的早熟,设变异概率为0.01。

(5)初始种群规模设为50,终止规则为连续20代没有更优的个体产生,即stall generations设为20;这样可加大候选种群的多样性,有利于找到全局最优。

3.2 算法设计

对于矩阵M,首先求其频繁1—项集,这是非常容易实现的,即对矩阵每列求内积即可,然后删除小于最小支持度min_sup的列(项),剩下的列就是频繁1—项集了。由于遗传算法两次得到的结果可能一样,因此再设置循环变量m,即在连续m次内没有新的最大频繁集生成即停止。

具体算法如下:

(1)按照映射f,将数据库D映射为矩阵M,将M调入内存中,并给出最小支持度min_sup。

(2)对矩阵M的各个列向量,分别求其内积,然后删除内积小于最小支持度min_sup的列,新的矩阵依然记为M。并分别置count = 0,$F = \varnothing$和给出循环变量m。

(3)设置循环,依次更新F,即最长频繁项集集合。伪码如下:

```
while count≤m
  x = run GA;
  if x in F
    count = count + 1;
  else F = F∪x;
    count = 0;
  end
end
```

(4)当结束条件满足时,F中的元素就是满足最小支持度min_sup的最长频繁项集。

4 实　验

文中对表1中的数据,用本法对其进行求解,得到2个最长频繁项集$\{I_1, I_2, I_3\}$,$\{I_1, I_2, I_5\}$,这与Apriori算法所得到的结果一致。

文中还随机产生了一组数据,共有2000个事务,20个项,编程环境为Matlab7.0。表2显示了在不同支持度下所得到的最长频繁项集的个数、长度、所用时间和循环次数。

表2　实验结果

支持度(%)	长模式集个数	长度	时间(s)	循环次数(m)
0.1	3	14	32.266	20
0.4	5	8	72.75	40
1	11	6	214.7	70
2	2	5	28.141	20

由表2可以看出,m的选取直接影响算法的运行效率,且如何选取最小m,使所有的最长频繁项集都能找到,这需要考虑,文中通过多次实验而得出最小的m。

最长频繁项集挖掘所需的时间并不随支持度的增加而减少,所需时间只与潜在的最长频繁项集的个数和循环次数有关,这正好与大多数算法支持度越大速度越快相反,文献[5]也得到了相同的结果。

出现上述现象主要是因为传统算法要找出每个候选集并还对其计数、筛选,这耗费了大部分时间。而遗传算法根据其全局寻找最优的特点,直接找到最长模式,因而节约了时间。在支持度较小时,最长模式集个数较少(但长度大),这也节约了时间,提高了效率。

5 结　语

文中提出基于遗传算法的长频繁集的挖掘算法,直接对矩阵进行操作,并且利用遗传算法的全局最优和海量数据搜索性,得出最长频繁项集。算法只需扫描一次数据库,能大大节约时间。实验也说明了该法的可行性。

参考文献:

[1] Han Jia Wei, Kamber M. 数据挖掘概念与技术[M]. 范　明等译. 北京:机械工业出版社,2001.

[2] 李　超,于昭平. 基于矩阵的Apriori算法改进[J]. 计算机工程,2001(23):68-69.

[3] 刘以安,羊　斌. 关联规则挖掘中对Apriori算法的一种改进研究[J]. 计算机应用,2007,27(2):418-420.

[4] 朱其祥,徐　勇,张　林. 基于改进Apriori算法的关联规则挖掘研究[J]. 计算机技术与发展,2006,16(7):102-104.

[5] 张泽洪,张　伟. 基于最长频繁闭项集的聚类算法[J]. 计算机工程,2007(1):187-192.

[6] 阎平凡,张长水. 人工神经网络与模拟进化计算[M]. 北京:清华大学出版社,2000:396-411.