文章编号:0253-2778(2007)02-0217-04

基于均值约束满足度剪枝策略的高效序列模式挖掘算法*

倪志伟[1],叶红云[1],曹欢欢[2]

(1.合肥工业大学管理学院,安徽合肥 230009;2.中国科学技术大学计算机科学与技术系,安徽合肥 230027)

摘要:为了减少无用候选序列的生成,并使挖掘得到的序列模式符合用户要求,约束条件下的频繁序列模式挖掘已成为数据挖掘领域的一个新的重要研究方向.作为强约束形式的一种,均值约束目前仍然是基于约束的频繁序列模式挖掘的一个困难问题,其主要原因在于很难利用均值约束来进行序列模式挖掘中的剪枝.为此,提出了一种基于均值约束满足度剪枝策略,并且以前缀增长方法为基础设计了一个有效的频繁序列模式挖掘算法.通过分析并实验验证了该算法的时间效率和剪枝性能,结果表明,该方法是有效的.

关键词:序列模式;均值约束;剪枝

中图分类号:TP181 **文献标识码**:A

Efficient sequential pattern mining algorithm based on average value constraint satisfaction pruning strategy

NI Zhi-wei[1], YE Hong-yun[1], CAO Huan-huan[2]

(1. *School of Management, Hefei University of Technology, Hefei* 230009, *China*;
2. *Department of Computer Science and Technology, University of Science and Technology of China, Hefei* 230027, *China*)

Abstract: To reduce the generation of useless candidates and make the generated patterns satisfy special requirements of users, constraint based on frequent sequential pattern mining has currently become an important research direction of data mining. However, as a kind of tough constraint, average value constraint is still a difficult problem to deal with because of its difficulty to be incorporated into the process of pruning candidates. An effective pruning strategy based on average value constraint satisfaction was proposed, and then a frequent sequential pattern mining algorithm was designed based on the prefix-growth method. In the end, the running efficiency and pruning performance of the proposed algorithm was analyzed by experiments. The results show that the proposed method is effective.

Key words: sequential pattern; average value constraint; pruning

研究快报

频繁序列模式挖掘是目前数据挖掘领域中的一个热点问题[1],在许多领域有着广泛的应用.针对频繁序列模式挖掘问题的主要工作有GSP算法[2],PrefixSpan算法[3]等.GSP算法通过自上而下、由短到长的方式生成候选序列来检测其频繁度达到发现频繁序列模式的目的,但当最小支持度过低时,算法将产生大量的候选序列模式而造成效率骤降;而PrefixSpan算法采用前缀增长的思想,有效地提高

* **收稿日期**:2006-11-14;**修回日期**:2006-11-30
基金项目:安徽省自然科学基金(050460402),安徽省教育厅科研项目(2006sk010)资助.
作者简介:倪志伟(通讯作者),男,博士/教授. E-mail:nzwgd@hfut.edu.cn

了效率.然而,满足最小支持度约束的频繁序列模式中,通常包含了大量用户并不感兴趣的模式.在实际应用中,通常以某种约束来表达用户感兴趣模式的类型,即用户常常只对满足某些特定约束的频繁子序列感兴趣.因此,近年来,研究利用约束,对用户不感兴趣的序列模式进行剪枝成为一个新的重要研究方向.目前的工作主要有两类,第一类针对某种约束,如针对间隔约束(gap)的CCSM[4]和针对单调约束(monotony)的ExAnte[5];第二类则是给出一个可以针对不同约束采取不同策略的算法框架,如PrefixGrowth[6]和CBPSAlgm[7].本文的主要工作就是结合一种通常认为难以和挖掘过程结合的约束——均值约束的特性,在CBPSAlgm算法框架基础上,提出一种高效的基于约束满足度的剪枝策略,设计了相应的处理平均值约束的新算法——MPAC(mining frequent sequential patterns with average value constraint),并对其剪枝策略的有效性和时间性能进行了验证.

均值约束要求序列中的每个项都有值(value)属性,约束是一个聚集(aggregate)函数,对于均值约束,约束函数 CF 表示为 Avg(Seq)≥min_Avg,或 Avg(Seq)≤max_Avg.其中,Avg(Seq)指计算序列 Seq 所有单项的平均 value.例如:一个市场分析员可能需要找出所有购买物品的序列模式中平均价值大于1 000元的模式.为便于更有针对性地说明问题,本文以 Avg(Seq)≥min_Avg 作为均值约束.在介绍挖掘策略和算法之前,首先定义如下几个概念.

定义 1 项 i 对均值约束的满足度表示为 Sat(i).若一个项的值大于 min_Avg,则 Sat(i)=i.value−min_Avg,否则,Sat(i)=0.

定义 2 序列 s 对均值约束的满足度表示为 Sat(s),是其包含的所有项对均值约束的满足度之和,即 $\mathrm{Sat}(s)=\sum_{j=0}^{|s|-1}\mathrm{Sat}(s[j])$.其中,|$s$|表示序列 s 的长度,$s[j]$ 表示序列 s 的第 j 个项.

定义 3 项 i 对均值约束的绝对满足度表示为 Abs_Sat(i).Abs_Sat(i)等于项 i 的值和 min_Avg 之差.

定义 4 序列 s 对均值约束的绝对满足度表示为 Abs_Sat(s),是其包含的所有项对均值约束的绝对满足度之和,即 $\mathrm{Abs\_Sat}(s)=\sum_{i=0}^{|s|-1}\mathrm{Abs\_Sat}(s[j])$.其中,|$s$|表示序列 s 的长度,$s[j]$ 表示序列 s 的第 j 个项.

显而易见,根据均值约束性质可知:序列 s 满足均值约束,当且仅当 s 对均值约束的绝对满足度 Sat(s)≥0.基于上述定义,可以设计出一个合理的项剪枝策略.该策略如下:

设序列 s_1 为前缀 Pre 的投影数据库 D(Pre)中的一个序列.若 s_1 中的项 i 是可被扩展的,当且仅当 Abs_Sat(Pre)+Abs_Sat(i)+Sat(currentsuffix)≥0.其中,currentsuffix 表示 s_1 中位于项 i 之后的后缀子序列.

为了高效地实现上述策略,关键在于为序列 s_1 中的每个项 i 构建相应的 Sat(currentsuffix),即序列中项 i 之后的后缀子序列对均值约束的满足度.构建方法如下:

(Ⅰ) 初始化 Sat(currentsuffix) = 0;

(Ⅱ) 从 s_1 的尾端向前逐个扫描序列中的每一项 i,若项 i 对均值约束的满足度 Sat(i)>0,则 Sat(currentsuffix)+=Sat(i).

(Ⅲ) 直至遍历完 s_1 中的所有项,并对遍历的每个项,按照上述项剪枝策略判断其是否可以进行剪枝.

本文在前缀增长方法的基础上,提出了基于均值约束的剪枝策略,并设计了相应的基于均值约束的频繁序列模式挖掘算法——MPAC 算法.下面首先对本文算法的主要思想进行简要描述,并在图 1、2 中给出了算法的具体描述.

step 1 算法扫描序列数据库,对于每一个序列 Seq 逆序扫描,如果单项 item 满足 Abs_Sat(item)+Sat(currentsuffix)≥0,则 item 是合法扩展项,算法对其进行计数.扫描完毕后,就得到所有合法的扩展项并输出满足约束函数 CF 的项.对于这些项,分别构造相应的投影数据库,将每一个项作为前缀,转到 step 2.

step 2 对于给定的前缀 Pre 及其投影数据库,逆序扫描投影数据库中的每一个 suffix,如果对于项 item,有 Abs_Sat(Pre)+Abs_Sat(item)+Sat(currentsuffix)≥0,则 item 是一个合法扩展项;如果 item 能作为新的元素与 Pre 合并,则算法对 item 的 Scounter 计数器计数;如果 item 能作为 Pre 最后一个元素的新项与 Pre 合并,则对 item 的 Icounter 计数器计数.扫描完毕后,用每一个频繁扩展项 item,对 Pre 进行扩展,得到新的前缀 Pre',如果 Pre' 满足 CF 则输出.然后,生成 Pre' 的投影数据

研究快报

库,并挖掘其中的频繁的合法扩展项,如此递归执行算法.图 1 给出了 MPAC 算法描述,图 2 给出了 MPAC 算法核心步骤 SubMPAC 的描述.

```
输入：序列数据库：SDB；最小支持度：min_support；均值约束 C
输出：满足 C 的频繁序列模式.
方法：call MPAC(SDB)；
    procedure MPAC (SDB, min_support, C,a)
    (1) for each suffix in SDB
    (2)   for each item in suffix
    (3)     if Abs_Sat(item) + Sat(currentsuffix) ≥ 0 then //
            合法扩展
    (4)       count item //计数
    (5)     end for
    (6) end for
    (7) add all frequent items to Set   //得到频繁的合法扩展
        单项
    (8) for each item in Set
    (9)     a' = Sextendprefix(a, item)   // item 作为新的
            element 扩展 a
    (10)    if CF(a') = true then out put a' //满足约束则
            输出
    (11)    call SubMPAC(a', SDB|a', min_support, C)
    (12) end for
    (13) end procedure
```

图 1　MPAC 的算法描述

Fig. 1　Algrithm description of MPAC

```
method SubMPAC (a, S|a, min_support, C)
(1) for each suffix in S|a
(2)   for each item i in suffix
(3)     currentsuffix = suffix/i   //以 i 为前缀的后缀序列
(4)     if Abs_Sat(a) + Abs_Sat(i) + Sat(currentsuffix) ≥ 0
(5)       if i can be a new element of a then
(6)         i.Scounter++;   // i.Scounter is initialized to be 0
(7)       if i can be a new item of a then
(8)         i.Icounter++;   // i.Icounter is initialized to be 0
(9)     end for
(10) end for
(11) for each item i
(12)   if i.Scounter > min_support then
(13)     a1 = Sextendprefix(a, i)   // i 作为新的 element 扩展 a
(14)     if CF(a1) = true then out put a1
(15)     call SubMPAC (a1, S| a1, min_support, C)
(16)   if i.Icounter > min_support then
(17)     a2 = Iextendprefix(a, i)   // i 作为新的 item 扩展 a
(18)     if CF(a2) = true then out put a2
(19)     call SubMPAC (a2, S| a2, min_support, C)
(20)   end for
(21) end method
```

图 2　SubMPAC 的算法描述

Fig. 2　Algrithm description of SubMPAC

为评估本文算法效率,我们利用 IBM 的数据生成程序(IBM data generator)产生的序列数据对算法的执行效率进行评测.序列数据集为 C10T5S4I2.5D50k,序列中每个 item 对应的 value 被设置为 itemNo(Mod)100+1.所有的实验是在 PIV 1.7G, 512 MB 内存的主机, Windows-XP professional 系统下完成.数据库系统为 Microsoft SQL server 2000.

图 3 显示了在支持度为 0.4%的情况下,均值剪枝策略的剪枝效率.纵坐标表示了剪枝策略在不同的约束下剪除搜索空间中不合法扩展项的百分比 pruning_ratio=[1-(约束条件下的频繁扩展项数)/(无约束时的频繁扩展项数)]*100%.如图 3 所示,随着约束的逐步加强,剪枝策略有效地将大量不合法的频繁扩展项剪除,从而大大提高了挖掘效率.

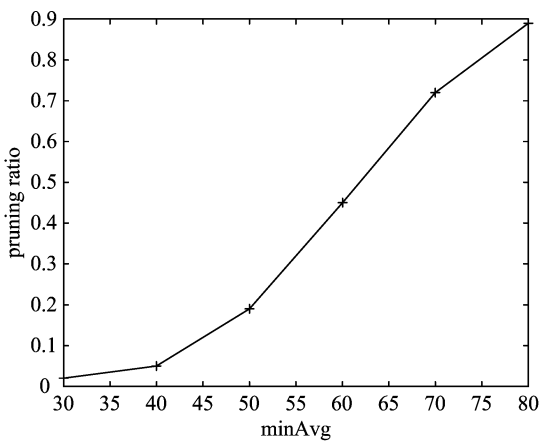

图 3　均值剪枝策略在支持度 0.4%下的剪枝效率

Fig. 3　Pruning efficiency of average value pruning strategy if support is 0.4%

图 4 展示了在各个不同的最小支持度下,不同均值约束下的 MPAC 算法的执行结果.数据集仍然采用 C10T5S4I2.5D50k 模拟序列数据库.从图中可以看出,随着均值约束逐渐增强,算法表现出较好的剪枝力度和执行效率.因此,实验结果表明,该策略对于均值约束是有效的.

本文针对一种典型的强约束——均值约束进行了研究.首先我们针对均值约束的特性提出了满足度的概念,然后基于这个概念设计了相应的剪枝策略,在此基础上,提出了基于均值约束的频繁序列模式挖掘算法——MPAC 算法.最后通过合成数据集上的实验,对剪枝策略进行了验证,结果表明该策略

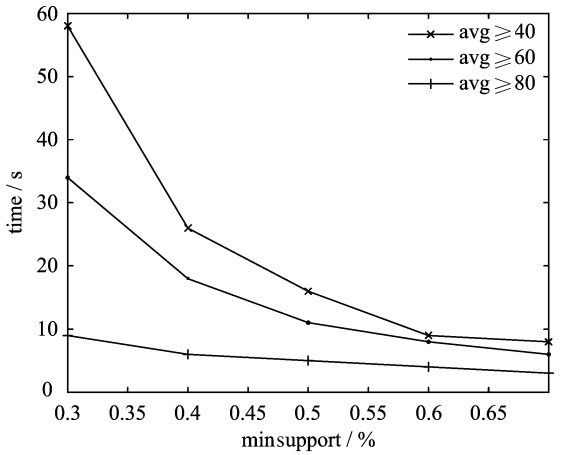

图 4 均值约束下的序列模式挖掘算法执行结果

Fig. 4 Running result of sequential pattern mining algorithm under average value constraints

是有效的,同时实验结果也表明本文提出的 MPAC 算法具有良好的时间性能.

参考文献(References)

[1] Wang J T L, Chirn G W, Marr T G, et al. Combinatorial pattern discovery for scientific data: some preliminary results [C]// Proc. of ACM SIGMOD Conference on Management of Data. Minneapolis, Minnesota, 1994: 115-125.

[2] Srikant R, Agrawal R. Mining sequential patterns: generalizations and performance improvements [C]// Proc. 5th Int. Conf. on Extending Database Technology. Avignon, France, 1996:3-17.

[3] Pei J, Han J, Mortazavi-Asl B, et al. PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth[C]// Proc. 2001 Int. Conf. Data Engineering. Heidelberg, Germany, 2001: 215-224.

[4] Orlando S, Perego R, Silvestri C. A new algorithm for gap constrained sequence mining [C]// ACM Symposium on Applied Computing. Nicosia, Cyprus, 2004:540-547.

[5] Bonchi F, Giannotti F, Mazzanti A, et al. ExAnte: anticipated data reduction in constrained patterns mining [C]// The 7th European Conference on Principles and Practice of Knowledge Discovery in Databases. 2003: 59-70, 56.

[6] Pei J, Han J, Wang W. Mining sequential patterns with constraints in large databases [C]// Proc. of CIKM Conference. McLean, VA, 2002:18-25.

[7] Chen E H, Li T S, Phillip C-y. Sheu: a general effective framework for monotony and tough constraint based sequential pattern mining[C]// Proc. of Data Warehouse and Knowledge Discovery. 2005: 458-467.

研究快报