第 38 卷 第 3 期
Vol.38 No.3

计 算 机 工 程
Computer Engineering

2012 年 2 月
February 2012

• 软件技术与数据库 •　　文章编号：1000—3428(2012)03—0031—03　　文献标识码：A　　中图分类号：TP311.1

基于动态项集计数的加权频繁项集算法

秦丽君[1,2]，罗雄飞[1]

(1. 中国科学院软件研究所，北京 100190；2. 中国科学院研究生院，北京 100190)

摘　要：基于 Apriori 的加权频繁项集挖掘算法存在扫描数据集次数多的问题。为此，提出一种基于动态项集计数的加权频繁项集算法。该算法采用权值键树的数据结构和动态项集计数的方法，满足向下闭合特性，并且动态生成候选频繁项集，从而减少扫描数据集的次数。实验结果证明，该算法生成的加权频繁项集具有较高的效率和时间性能。

关键词：数据挖掘；加权频繁项集挖掘；动态项集计数；加权支持度；权值键树；向下闭合特性；最大权值

Weighted Frequent Itemset Algorithm Based on Dynamic Itemset Counting

QIN Li-jun[1,2], LUO Xiong-fei[1]

(1. Institute of Software, Chinese Academy of Sciences, Beijing 100190, China;
2. Graduate University of Chinese Academy of Sciences, Beijing 100190, China)

【Abstract】The existing weighted frequent itemset mining algorithms which are based on Apriori require multiple dataset scans. This paper proposes a weighted frequent itemset algorithm weighted frequent itemset mining based on dynamic itemset counting which uses the structure of weighted trie tree and the method of dynamic itemset counting. This algorithm satisfies the downward closure property and dynamically generates candidate frequent itemsets, thereby reduces the number of scanning datasets and improves the performance. Experimental results show that the proposed algorithm not only generates the weighted frequent itemsets, but also has high efficiency and time performance.

【Key words】data mining; weighted frequent itemset mining; dynamic itemset counting; weighted support degree; weighted trie tree; downward closure property; maximum weight

DOI: 10.3969/j.issn.1000-3428.2012.03.011

1　概述

加权频繁项集挖掘是数据挖掘中的一个重要研究课题，相比于传统频繁项集挖掘，它能发现那些出现频率较低但权值比较大的重要频繁项集。已有的加权频繁项集算法 WAR[1] 和 WARM[2]在扫描完整遍数据集后生成候选频繁项集，需要多遍扫描数据集。本文提出一种权值键树结构，用于保存生成的频繁项集和一种基于动态项集计数的加权频繁项集算法，该算法在扫描完整遍数据集前多次生成候选频繁项集，相比于 WAR 算法和 WARM 算法，所需扫描数据集的次数少，从而提高时间性能。

2　相关概念

事务数据集中的每个属性项都有一个权值，代表了此项的重要程度，该权值是一个非负的实数[1-5]。项集的权值为其中所有属性项权值的平均值[3-4]。如在项集 $P=\{x_1,x_2,\cdots,x_m\}$ 中，$x_k(1\leqslant k\leqslant m)$ 是一个属性项，其权值为 $W(x_k)$，m 是 P 中属性项的个数，则 P 的权值为：

$$W(P)=\sum_{k=1}^{k=m}W(x_k)/m \tag{1}$$

设 P 中没有考虑权值的支持度为 $\sup(P)$。在非加权频繁项集挖掘中，当 P 的支持度大于或等于最小支持度 s 时，它是一个频繁项集。在加权频繁项集挖掘中 P 的加权支持度为：

$$W\sup(P)=\sup(P)\times W(P) \tag{2}$$

如果一个项集的加权支持度不小于 s，则此项集是加权频繁项集。

非加权频繁项集算法具有向下闭合特性(Downward Closure Property)[1,6-7]，即频繁项集的子集都是频繁项集。使用等式(2)计算得到加权频繁项集不满足向下闭合特性。IWFPWA[3]和 WFIM[4]提出项集的最大权值，它是数据集中所有属性项的最大权值，缩写为 MAW。设 n 为总共的属性项的个数，P 的最大权值的计算公式为：

$$\mathrm{MAW}(P)=\{y\mid y=\max(W(x_k)),1\leqslant k\leqslant n\} \tag{3}$$

P 的最大权值支持度为：

$$\mathrm{MAW}\sup(P)=\sup(P)\times\mathrm{MAW}(P) \tag{4}$$

MAW 计算的最大权值支持度满足向下闭合特性。

3　本文工作

3.1　数据结构与算法描述

算法采用的数据结构为权值键树，树中结点有不同的状态，结点的形状代表其不同的状态。本文在 DIC[6]已有的 4 种描述项集状态的基础上再添加 2 种状态，它们的描述如表 1 所示。表 2 和表 3 给出了一个带有权值的事务数据集，数据集中有 {a,b,c,d,e} 5 个属性项，表 2 中有 8 条记录，表 3 给出属性项的权值，最小支持度为 $s=0.29$。

基金项目：国家“863”计划基金资助项目(2007AA040702)

作者简介：秦丽君(1983－)，女，硕士研究生，主研方向：数据挖掘，数据可视化；罗雄飞，博士

收稿日期：2011-06-29　**E-mail：**weiweiqueena@gmail.com

表 1 状态符号定义

状态	符号	定义
实方形	SS	候选频繁项集结点已经扫描完整遍数据集，且加权支持度不小于 s，是频繁项集结点
实环形	SC	候选频繁项集结点已经扫描完整遍数据集，且最大权值支持度小于 s，是非频繁项集结点
实菱形	SR	候选频繁项集结点已经扫描完整遍数据集，且最大权值支持度不小于 s，加权支持度小于 s
虚方形	DS	候选频繁项集结点未扫描完整遍数据集，且加权支持度不小于 s，是候选频繁项集结点
虚环形	DC	候选频繁项集结点未扫描完整遍数据集，且最大权值支持度小于 s，是候选非频繁项集结点
虚菱形	DR	候选频繁项集结点未扫描完整遍数据集，且最大权值支持度不小于 s，加权支持度小于 s

表 2 事务数据集

TID	事务
T1	a, b, c, d
T2	a, c, d, e
T3	a, b, c, d
T4	a, c, d
T5	a, b, c, d
T6	a, c, d
T7	a, c, d, e
T8	a, b, c, d, e

表 3 属性项权值

Item	Weight
a	0.8
b	0.6
c	0.5
d	0.4
e	0.6

在本文提出的权值键树数据结构中，每个项集结点的维护项有属性 ID、计数器、权值和最大权值等属性，分别表示此结点代表的项集、项集在数据集中出现的次数、项集权值和最大权值，且每个结点有 6 种状态，如表 1 所示。图 1~图 5 描述了本文使用表 2、表 3 的数据生成的权值键树过程。项集结点的属性之间用分隔符“/”隔开。

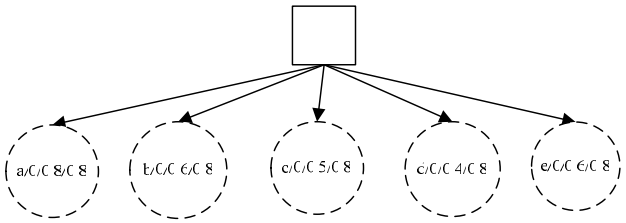

图 1 WDIC 算法初始结构

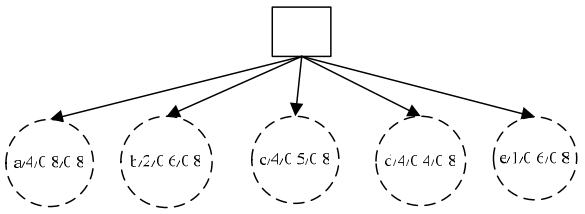

图 2 扫描 M 条记录进行计数的结构

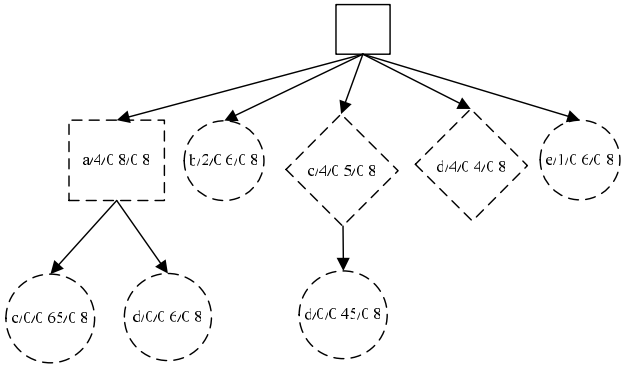

图 3 M 条记录后生成候选频繁项集结点的结构

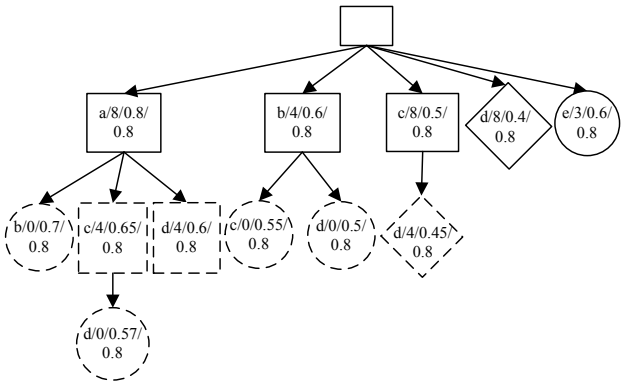

图 4 扫描完 1 遍数据集后的结构

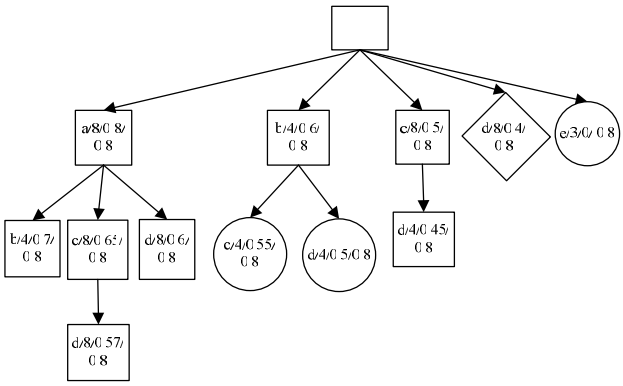

图 5 扫描完 2 遍数据集后的结构

本文提出的基于动态项集计数的加权频繁项集算法 WDIC 包括以下 4 个步骤：

Step1 生成空根结点，状态为 SS。生成数据集的所有属性项集结点，结点状态为 DC，初始化计数器为 0，使用等式(1)计算结点的权值和使用等式(3)计算最大权值，得到权值键树 Tree 的初始结构，如图 1 所示。

Step2 在 Tree 上，扫描 M(M 小于数据集总记录个数)条记录，改变 Tree 中结点的计数器。在扫描每条记录时都对 Tree 深度优先遍历，对在记录中的结点的计数器加 1，如图 2 所示。当扫描到数据集末尾时，把扫描点重新定位到数据集头。

Step3 扫描完 M 条记录后，改变结点状态，并生成新的候选频繁项集结点。深度优先遍历 Tree，根据表 1 改变结点状态，并生成新结点，新结点的子项集的最大权值支持度不小于 s，如图 3~图 5 所示。

Step4 重复地从 Step2 开始执行，直到 Tree 中所有结点状态为实形，即 SC、SR 或 SS，算法结束。

3.2 数据结构与算法实例分析

利用表 2 和表 3 的事务数据集挖掘加权频繁项集，设 M 为 4，如图 1~图 5 所示。

图 1 为权值键树初始化的结构，根节点下有 5 个属性项子结点，它们的计数为 0，计算它们权值和最大权值，每个结点的状态都是 DC。

图 2 为扫描完 M=4 条记录并对权值键树计数的结构，如 a 在前 4 条记录中都出现了，则其计数为 4，e 的计数为 1。

图 3 是对图 2 计数后结点状态改变，结点 a 的加权支持度为 $4/8\times0.8=0.4>0.29$，其状态为 DC。c 的最大权值支持度为 $4/8\times0.8=0.4>0.29$，加权支持度为 $4/8\times0.5=0.25<0.29$，其状态为 DR。e 的最大权值支持度为 $1/8\times0.8=0.1<0.29$，其状态仍为 DC。由于 a , c , d 的状态为方形或菱形，因此生

成它们的下级集结点，a 结点下有 c 结点和 d 子结点，c 结点下有 d 子结点。

图 4 为第 2 次扫描完 $M=4$ 条记录更新权值键树中的结点并改变状态和生成新的子结点的结果，此时已经扫描完一遍数据集，所有的一项集结点都扫描完整遍数据集，它们的状态需变成实形。

图 5 是生成的最终权值键树。该算法只扫描了 2 遍数据集，最大的频繁项集为 {a,c,d} 。因为 {a,c,d} 中有 3 个属性项，利用基于 Apriori[7]的算法需要扫描 3 遍数据集。相比而言，本文算法减少了扫描数据集次数，从而提高了算法的性能。

4　实验结果与分析

本节对 WDIC 生成重要频繁项集个数和其性能进行了实验，分别与非加权的频繁项集算法(表示为 DIC)和基于 Apriori 的加权频繁项集算法(表示为 WFI)进行对比。在实验中使用的数据集为大型数据挖掘工具 SPSS Clementine[8]的 transaction 文件中的测试数据，它是一超市中的购物数据，其中，有 20 个属性项，设置 M 为数据集个数的 1/5。实验环境是 Pentium(R) 2.80 GHz 处理器，2 GB 内存，Windows XP 操作系统的台式电脑。

4.1　重要频繁项集的生成实验

图 6 给出了最小支持度对生成重要频繁项集个数影响的实验结果，数据集记录为 4.5 万条，重要项个数为 3，最大权值为 3。从图中可以看出，随着最小支持度大小逐步减少，WDIC 和 DIC 生成的重要频繁项集个数增多，但 WDIC 生成个数的增加幅度明显高于 DIC，因为最小支持度越小，生成的频繁项集越多，重要项的频繁项集相对越多。

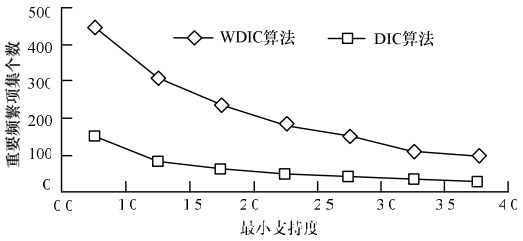

图 6　最小支持度与重要频繁项集个数的关系

4.2　算法性能实验

图 7 给出了数据集记录个数对性能影响的实验结果。重要项个数为 3，最大权值为 3，最小支持度为 0.2。由图可知，随着数据集记录越来越多，算法性能逐渐变差，但 WDIC 性能优于 WFI。随着数据集记录的增加，WFI 性能变差的幅度明显高于 WDIC，因为 WDIC 比 WFI 扫描数据集次数少，随着数据集记录的增加，扫描记录的总个数相对更少。

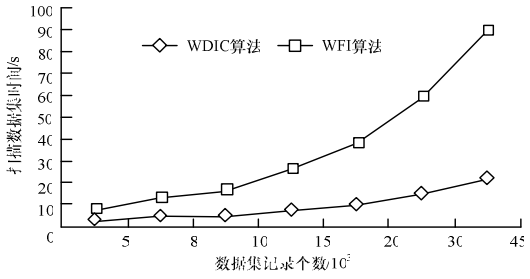

图 7　数据集记录个数与时间性能的关系

5　结束语

本文提出一种基于动态项集计数的加权频繁项算法，相比于经典加权频繁项集算法，它扫描数据集的次数较少，从而提高了算法的时间性能，因为现实数据中数据是不断增加改变的。下一步的工作将算法扩展到动态加权频繁项挖掘中。

参考文献

[1] Wang Wei, Yang Jiong, Yu P S. WAR: Weighted Association Rules for Item Intensities[J]. Knowledge Information and Systems, 2004, 6(2): 203-229.

[2] Tao F. Weighted Association Rule Mining Using Weighted Support and Significant Framework[C]//Proc. of the 9th ACM SIGKDD'03. [S. 1.]: ACM Press, 2003: 661-666.

[3] Ahmed C, Taneer S, Jeong B, et al. Mining Weighted Frequent Patterns in Incremental Databases[C]//Proc. of PRICAI'08. Hanoi, Vietnam: [s. n.], 2008: 933-938.

[4] Yun U, Leggett J J. WFIM: Weighted Frequent Itemset Mining with a Weight Range and a Minimum Weight[C]//Proc. of the 4th SIAM Int'1 Conf. on Data Mining. College Station, USA: [s. n.], 2005: 636-640.

[5] 傅国强，郭向勇. 动态加权关联规则算法的分析与实现[J]. 计算机工程, 2010, 36(23): 79-81.

[6] Brin S, Motwani R, Ullman J, et al. Dynamic Itemset Counting and Implication Rules for Market Basket Data[C]//Proc. of 1997 ACM SIGMOD Int'1 Conf. on Management of Data. [S. 1.]: ACM Press, 1997: 255-264.

[7] Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules[C]//Proc. of the 20th Int'1 Conf. on Very Large Data Bases. Santiago, Chile: [s. n.], 1994: 487-499.

[8] SPSS. Data Mining(Clementine)[DB/OL]. (2008-10-30). http://www.spss.com/data mining/.

编辑　索书志

(上接第 24 页)

参考文献

[1] Kanda M. Practical Security Evaluation Against Differential and Linear Attacks for Feistel Ciphers with SPN Round Function[C]// Proc. of Selected Areas in Cryptography. New York, USA: Springer-Verlag, 2000: 158-179.

[2] 师国栋，康　绯，顾海文. 分组密码统一描述模型研究[J]. 计算机工程, 2010, 36(1): 154-156.

[3] 刘连浩，罗　安，陈松乔. 基于十进制的加密技术研究[J]. 小型微型计算机系统, 2006, 27(7): 1229-1231.

[4] Knudsen L R. Practically Secure Feistel Ciphers[C]//Proc. of Lecture Notes in Computer Science. New York, USA: Springer-Verlag, 1994: 211-221.

[5] 高靖哲，赵新杰，矫文成，等. 针对 CLEFIA 的多字节差分故障分析[J]. 计算机工程, 2010, 36(19): 156-158.

[6] 吴文玲，贺也平. 一类广义 Feistel 密码的安全性评估[J]. 电子与信息学报, 2002, 24(9): 1177-1184.

[7] 李　超，屈龙江，李　强. 对 DES 的一种新的线性分析[J]. 国防科学技术大学学报, 2004, 26(3): 43-47.

[8] 张　鹏，孙　兵，李　超. 对特殊类型 Feistel 密码的 Square 攻击[J]. 国防科学技术大学学报, 2010, 32(4): 137-141.

[9] 刘连浩，崔　杰，刘上力. 一种 AES S 盒改进方案的设计[J]. 中南大学学报：自然科学版, 2007, 38(2): 339-344.

[10] Daemen J, Rijmen V. AES Proposal: Rijndael, Version2[EB/OL]. (1999-07-10). http://www.esat.kuleuvan.ac.be/~rijndael.

编辑　顾逸斐