基于多最小支持度的关联规则挖掘①

晏 杰[1], 亓文娟[2], 郭 磊[2], 黄书城[2]

[1](武夷学院 团委, 武夷山 354300)
[2](武夷学院 数学与计算机学院, 武夷山 354300)

摘 要: 分析了单最小支持度关联规则挖掘的局限性, 提出了基于多最小支持度的关联规则挖掘模型, 重点研究了多最小支持度MS-Apriori算法的基本思想, 指出了算法的优缺点并通过实例说明发现频繁项集的方法, 最后指出该算法的不足及改进算法.

关键词: 关联规则; 多最小支持度; MS-Apriori 算法

Based on Multiple Minimum Supports of Association Rules in Data Mining

YAN Jie[1], QI Wen-Juan[2], GUO Lei[2], HUANG Shu-Cheng[2]

[1](Youth League Committee ,Wuyi University, Wuyishan 354300, China)
[2](Mathematics and Computer Science college,Wuyi University, Wuyishan 354300, China)

Abstract: This paper analyzed the limitation of the single minimum support degree of association rule mining. We put forward the model of association rule mining based on multiple minimum support. Our work focus on the multiple minimum support MS - Apriori algorithm the basic idea and points out the advantages and disadvantages of the algorithm with practical examples. Our work found that the method of frequent itemsets, and finally points out the shortage of the algorithm and improved algorithm.

Key words: association rules; multiple minimum supports; MS-Apriori algorithm

关联规则挖掘在数据挖掘中占有极其重要的地位, 最早是由 R.Agrawal 等人在 1993 年提出来的, 是用于发现隐藏在大型数据集中令人感兴趣的联系, 即发现数据项集之间潜在的关联或依赖联系. 为了发现有意义的规则, 需要给定两个阈值: 最小支持度和最小置信度. 传统的关联规则挖掘大多采用单最小支持度, 本文针对频繁项集发现算法使用单支持度的不足进行了分析, 提出了多支持度的关联规则挖掘算法 MS-Apriori, 研究了该算法的基本思想, 并通过具体的实例说明发现频繁项集的方法, 同时指出了算法的不足及改进算法, 旨在对关联规则挖掘算法的扩展和改进奠定基础.

1 单支持度挖掘的局限性

Apriori 算法是挖掘布尔关联规则频繁项集的经典算法, 围绕着该算法出现了不少改进算法, 比如不产生候选项集算法 FP-growth 算法[1], 它克服了 Apriori 算法中多次扫描事务数据库的缺陷, 只需对事务数据库进行二次扫描, 将发现长频繁模式的问题转化递归模式增长的策略, 避免产生的大量候选集, 大大降低了算法的时间复杂度. 但是这些改进算法只设定一个最小支持度阈值来限制需要的搜索空间及规则产生的数量, 其中隐含着这样一个假设: 数据集中各项具有相同或相似的出现频率, 然而现实情况并非如此, 如果采用单支持度进行关联规则挖掘, 将会遇到以下两个问题:

(1) 如果将最小支持度阈值设置的过高, 则在挖掘频繁项集的过程中, 由于出现频率较低项的支持度低于最小支持度阈值而被过滤掉, 但在实际生活中, 我们往往更关注包含出现频率较低项的规则, 它有可能

① 收稿时间:2013-04-28;收到修改稿时间:2013-09-22

© 中国科学院软件研究所 http://www.c-s-a.org.cn

会给我们带来价值. 比如商场奢侈品的购买频率比日常生活用品小, 由于奢侈品的利润高, 它的购买模式对于商场来说非常重要.

(2) 如果将最小支持度阈值设置的过低, 这样就会导致组合爆炸, 符合要求的频繁项集和关联规则的数目将以指数级的速度增长, 严重降低了算法的效率, 产生大量的无实际意义的关联规则, 同时用户要从大量的关联规则中找出有价值的规则, 必然增加了关联规则评价的难度.

以上问题被称为稀有项问题.

2 多支持度的关联规则挖掘

2.1 多支持度扩展模型

通常的挖掘策略都是为了涵盖发生概率较大的事务而舍弃稀有项[2]. 为了解决稀有项问题, Liu 等学者提出了一种多支持度关联规则挖掘模型 MIN 模型和基于该模型的 MS-Apriori 算法. 在 MIN 模型中, 关联规则的定义和单支持度模型是一致的, 但最小支持度的定义发生了改变, 即数据库中每个项目有一个最小项支持度, 用户可以根据项目出现的频率为每个项目指定不同的最小项支持度阈值.

定义 1: 最小项支持度[3]

对于事务数据库 D 的数据项集 I={i1, i2, …in}, 对任意项 $i \subset I$, 赋予 i 所需满足的最小支持度, 称之为最小项支持度(minimum item support), 记作 MIS(i).

定义 2: 最小支持度

对于项集 A={ i1, i2, …ik }($1 \leqslant k \leqslant n$), 其最小支持度为 MS(A)=min(MIS(i1), MIS(i2), …MIS(ik)), 即项集 A 的最小支持度为所包含各个项的最小项支持度的最小值.

定义 3: 多最小支持度下的频繁项集

若任意项 i 的支持度 support(i)≥MIS(i),则 i 是频繁的, 若项集 A 的支持度 support(A)≥MS(A),则 A 是频繁项集.

由以上定义可以得出, 对于出现频率较高的数据项组成的关联规则有较高的最小支持度限制, 相反则有较低的最小支持度限制. 对于任意候选规则的最小支持度都由其本身所需满足的最小支持度决定, 解决了关联规则挖掘中的稀有项问题.

2.2 MS-Apriori 算法

MS-Apriori 算法是采用多最小支持度进行关联规则挖掘的算法, 该算法是 Apriori 算法的改进算法. Apriori 算法使用 Apriori 性质[4](频繁项集的所有非空子集也是频繁的)来生成候选项集, 对于多最小支持度下该性质不再成立. 例如: 在事务数据库中有 4 个项目 A、B、C、D,它们的最小项支持度分别是: MIS(A)=0.1, MIS(B)=0.2, MIS(C)=0.05, MIS(D)=0.06, 如果 2-项集{A,B}的支持度是 0.09, 因为 sup{A,B}< min(MIS(A), MIS(B))=0.1,所以项集{A,B}将被删除, 这样就不会产生潜在的 3-项目集{A,B,C},{A,B,D}, 因为有可能 sup{A,B,C} ≥ min(MIS(A), MIS(B) , MIS(C))=0.05, sup{A,B,D}≥min(MIS(A), MIS(B) , MIS(D))=0.06,因此过早的删除项集{A,B}是错误的, 反之, 如果不删除项集{A,B}, Apriori 算法的向下封闭性将失效. 通过在挖掘过程中将每个事务的所有项按最小项支持度进行升序排列来解决,以此来满足类似于 Apriori 性质的向下封闭性, 然后采用类 Apriori 算法逐层搜索的迭代方法产生频繁项集.

MS-Apriori 算法伪代码如下:

输入: 事务数据库 D, 各项的最小支持度集 MIS

输出: D 中的频繁项集 L

step1: M=sort(MIS,I);// 根据 MIS 对项集 I 进行升序排列

step2: F=init-pass(M,D); // 进行初次扫描数据库

step3: L1={f | f∈F,f.count≥MIS(f)}; // 根据 MIS, 产生频繁 1-项集

step4: for(k=2;Lk-1≠∅;k++){

step5: if(k=2)

step6: C2=level2-candidate-gen(F) // 产生候选 2-项集

step7: else

step8: Ck=candidate-gen(Lk-1) // 根据 LK-1 产生候选 k-项集

step9: for each transaction t ∈D{

step10: Ct=Subset(Ck,t);// 得到 t 的子集为候选

step11: for each candidate c ∈Ct

step12: c.count=c.count+1;}// 计算每个候选 c 的支持度计数

step13: Lk={ c∈Ck | c.count≥MIS(c[1])}; // 产生频繁 k-项集

step14: }

step15: return L=L∪Lk

© 中国科学院软件研究所 http://www.c-s-a.org.cn

2.3 算法举例

设事务数据库 D,各项最小支持度为 MIS(A)=MIS(B)=0.4, MIS(C)=MIS(D)=MIS(F)=0.3, MIS(E)=0.2, 如图 1 所示.

TID	Items
T1	A,B,C,E
T2	B,D,E
T3	B,C
T4	A,B,D,F
T5	A,C,F
T6	A,C,D
T7	A,D,E
T8	A,B,E
T9	A,C,D,E
T10	B,C,D

按照 MIS 值升序排列

TID	Items
T1	E,C,A,B
T2	E,D,B
T3	C,B
T4	D,F,A,B
T5	C,F,A
T6	C,D,A
T7	E,D,A
T8	E,A,B
T9	E,C,D,A
T10	C,D,B

图 1 排序后的事务数据库 D

第 1 步: 根据 MIS 值对项集 I 排序 M={E, C, D, F, A, B}.

第 2 步: 扫描数据库 D,得到 M 中各项的实际支持度计数 E=0.5, C=0.6, D=0.6, F=0.2, A=0.7, B=0.6. 查找满足最小支持度的数据项 i,因为 sup(E)≥MIS(E)=0.2, 则加入到 F 中, 对 M 中 i 的每个后继项目 j, 如果 sup(j)≥MIS(i), 则将 j 也加入到 F 中, 因此 F={E, C, D, F, A, B}.

第 3 步: 对于 F 中的每项 i, 如果 sup(i)≥MIS(i), 就加入到 L1 中, 得到频繁 1-项集 L1={{E}, {C}, {D}, {A}, {B}}, 由于 sup(F)=0.2<MIS(F)=0.3, 所以 L1 中不含 F.

第 4 步至第 8 步: 候选项集的产生过程. 与传统 Apriori 算法生成候选 2-项集剪枝不同, 生成 C2 的方法比较特殊, 根据定理对于任意频繁 2-项集 A=<a, b>, 必有 sup(a)≥MIS(a)且 sup(b)≥MIS(a)来产生 C2. 长度大于 2 的候选项集的产生过程同 Apriori 算法类似, 包含连接和剪枝两步, 只是剪枝增加了限制条件. C2 各项及支持度分别为{{AB 0.3}, {AC 0.4}, {AD 0.4}, {AE 0.4}, {BC 0.3}, {BD 0.3}, {BE 0.3}, {CD 0.3}, {CE 0.2}, {DE 0.3}, {EF 0}}. C3 各项及支持度分别为{{ABC 0.1}, {ABD 0.1}, {ABE 0.2}, {ACD 0.2}, {ACE 0.2}, {ADE 0.2}, {BCD 0.1}, {BCE 0.1}, {BDE 0.1}, {CDE 0.1}}.

第 9 步到第 14 步: 扫描数据库 D, 根据候选项集中数据项的支持度与最小支持度进行比较, 确定频繁项集. L2={{AC}, {AD}, {AE}, {BC}, {BD}, {BE}, {CD}, {CE}, {DE} }, L3={{ABE}, {ACE}, {ADE}}.

第 15 步: 合并各长度的频繁项集得到所有的频繁项集. L={{E}, {C}, {D}, {A}, {B} , {AC} , {AD} , {AE}, {BC}, {BD}, {BE}, {CD}, {CE}, {DE}, {ABE}, {ACE}, {ADE} }.

算法最终得到的频繁项集和 Apriori 算法是一样的, 但用 MS-Apriori 算法减少了搜索空间, 节省了时间消耗.

2.4 算法不足及改进

MS-Apriori 算法虽然解决了单支持度无法挖掘稀有项之间的关联规则问题, 但由于限制条件过于宽松导致生成的关联规则数量过多, 同时 MIS 值的设置方式也可能影响关联规则的可用性.

文献[5]提出了一种基于概率的多最小支持度关联规则算法, 该算法有效的挖掘出发生概率较低事件中的关联规则, 但由于降低了概率较低项的支持度, 存在着候选项集增多的缺点. 文献[6]提出了使用相关项目集中各项目最小支持度中的最大值来实现剪枝, 这样使得最大频繁项集的生成过程和 Aprior 算法基本类似, 避免产生过多无实际意义的规则, 同时能够挖掘发生频率较低的规则, 算法效率大大提高. 文献[7]提出了一个基于多最小支持度的加权关联规则挖掘算法, 允许用户为每个数据项设置不同的权重和最小支持度, 从而解决数据项重要项不同, 出现频率不同的问题. 但是由于采用的是类 Apriori 算法, 仍然存在着产生大量的候选项集, 重复扫描事务数据库 I/O 开销很大等不足, 同时挖掘仅针对正关联规则的挖掘, 没有对负关联规则进行挖掘, 在挖掘过程中只注重频繁项集的的挖掘, 忽视了非频繁项集的重要作用.

3 总结

针对传统的关联规则采用单支持度进行挖掘, 不能解决稀有项问题, 本文提出了多支持度的关联规则挖掘算法 MS-Apriori, 研究了该算法的基本思想, 并通过具体的实例说明发现频繁项集的方法, 同时指出了算法的不足及改进算法. 针对关联规则的挖掘, 有待进一步研究.

参考文献

1 晏杰,亓文娟.基于 Aprior & FP-growth 算法的研究.计算机

(下转第 219 页)

© 中国科学院软件研究所 http://www.c-s-a.org.cn

3 实际应用

为了验证，算法的有效性与普遍实用性，选取南京航空航天大学的学生食堂作为实际验证的场所. 南京航空航天大学的食堂放菜的餐盘均为圆型，但是直径与颜色各有差异，目前是以大小来区分不同的价格.

为了能够通过识别菜盘特征以达到识别不同食物的过程，在实际应用中，对每一个盘子分配一种菜品，每种菜品与每种类型的盘子之间一一对应，首先选择菜品，放入托盘中，令托盘置于识别摄像头下，图象发送至计算机，通过本方案配套的软件识别出餐盘的种类以及数量，进而识别出所消费的菜品种类以及数量，通过查询内置菜品对应价目表，自动累加得到总价格，该价格显示在液晶屏幕上，若消费者对价格无异议，便可刷卡通过.

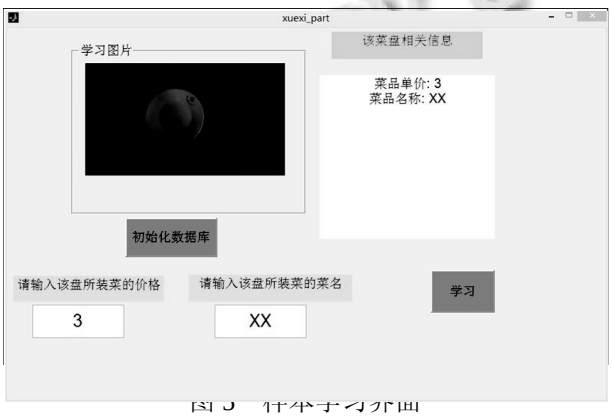

图 5 样本学习界面

4 结论

本文提出一种基于菜盘特征提取的食堂自动支付系统的设计方案. 该方案通过对餐盘的特征的提取和识别，成功实现了自动识别计算价格、自助缴费的食堂自动支付过程. 通过实际的实验验证，系统识别成功率较高，有相当实际应用价值. 但是，对于餐盘边缘破损的问题的导致图象特征丢失，对识别率影响比较明显. 与此同时，该系统的应用不需要改造食堂现有的餐盘，相比于采用 RFID 技术，在成本上优势明显，有很强的应用与推广价值.

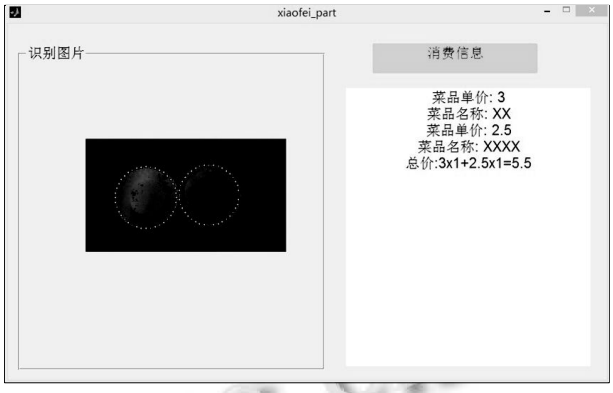

图 6 自动识别界面

参考文献

1 Mark S. Nixon. Feature Extraction and Image Processing. 2nd ed. New York: Elsevier, 2008: 115–148.

2 王慧燕.图像边缘检测和图像匹配研究及应用[学位论文].杭州:浙江大学,2003.

3 林开颜,吴军辉,徐立鸿.彩色图像分割方法综述.中国图象图形学报,2005,(1):1–10.

4 宗节保,段柳云,王莹,等.基于 MATLAB GUI 软件制作方法的研究与实现.电子设计工程,2010,(7):54–56..

5 段瑞玲,李庆祥,李玉和.图像边缘检测方法研究综述.光学技术,2005(3):415–419.

6 刘鹏宇.基于内容的图像特征提取算法的研究[学位论文].长春:吉林大学,2004.

(上接第 239 页)

系统应用,2013,22(5):122–125.

2 王瑄.多最小支持度下的关联规则研究[硕士学位论文].长春:长春理工大学,2008.

3 宋蓓.面向零售数据的关联规则挖掘算法的研究与应用[硕士学位论文].青岛:青岛科技大学,2009.

4 蒋盛益,李霞,郑琪.数据挖掘原理与实践.北京:电子工业出版社,2011.

5 田启明,王丽珍,尹群.一种基于概率的多最小支持度挖掘算法.计算机仿真,2006,7:115–118.

6 何朝阳,赵剑锋,江水.最大值控制的多最小支持度关联规则挖掘算法,2006,6:103–105.

7 邹力鹗,张其善.基于多最小支持度的加权关联规则挖掘算法. 北京航空航天大学学报,2007,33(5):590–593.

© 中国科学院软件研究所 http://www.c-s-a.org.cn