

基于语义划分的多层关联规则冗余处理方法

李帆, 夏士雄, 张磊, 王志愿

(中国矿业大学 计算机科学与技术学院, 江苏 徐州 221116)

摘 要: 针对多层次关联规则的冗余问题, 提出基于语义划分的多层关联规则冗余处理方法. 该方法根据语义描述信息将冗余规则划分为三类: 子规则冗余、同义规则冗余、可规约规则冗余. 冗余处理的时候删掉冗余子规则, 对同义规则进行归类视为具有相同意义的规则, 把只表述了部分信息的可归约规则规约到更高层次. 使用某手机订阅服务的真实数据进行实验, 实验表明该方法能有效地对规则进行冗余处理.

关键词: 多层关联规则; 冗余处理; 本体; 语义划分

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2010)08-0252-04

An Approach Based on Semantic Dividing for Eliminating Multi-level Redundant Association Rules

LI Fan, XIA Shi-Xiong, ZHANG Lei, WANG Zhi-Yuan

(School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)

Abstract: In this paper, aiming at the redundant rules produced during multi-level association rule mining, an approach for eliminating multi-level redundant rules based on semantic dividing is proposed. In the method, the redundancies are divided combination with semantic ontology into three categories: sub-rules redundancies, synonymous redundancies and reducible redundancies. Deleting the redundant sub-rules, grouping the synonymous rules and treating these rules having the same meaning, reducing rules that express only part of information to higher level, when pruning redundancies. Finally, the approach is evaluated on a real mobile subscription service dataset, and its effective in treatment of the redundant rules is proved through the experiment.

Key words: multi-level association rule; redundancy treatment; ontology; semantic dividing

1 引言

关联规则冗余处理主要有两类方法: (1) 挖掘过程中针对规则生成过程或规则本身进行冗余处理^[1-3]. 这类方法对在具有层次特征数据集上的关联规则挖掘冗余处理不能达到很好的效果; (2) 利用领域知识处理冗余规则^[4-9]. 这类方法利用语义本体表达领域知识, 使本体中概念或者属性对应关联规则中的项, 在关联规则挖掘完成后进行基于先验知识的冗余处理. 本文结合领域本体, 针对在具有层次特性数据库上的关联规则挖掘进行冗余处理.

本体用来描述领域信息, 本体节点表示领域概念或概念属性, 属性或概念间的连接表示它们之间的关系. 文中在关联规则挖掘基础上讨论基于领域本体的多层次关联规则挖掘过程中产生的冗余, 结合语义把产生的冗余规则划分为三类: 子规则冗余、同义规则冗余、可规约规则冗余, 冗余处理的时候删掉冗余子规则, 对同义规则进行归类视为具有相同意义的规则, 把只表述了部分信息的可归约规则规约到更高层次.

收稿日期: 2010-05-13; **修回日期:** 2010-06-21

基金项目: 国家自然科学基金项目(50674086); 高等学校博士学科点专项科研基金(20060290508); 江苏省博士后基金(0802023C); 中国矿业大学青年基金(2008A40)

2 关联规则的语义划分

设 $I = \{I_1, I_2, \dots, I_m\}$ 是全体数据项集合,其中每个项 $I_j (1 \leq j \leq m)$ 均为本体中的概念或属性. 数据项集 X 是由数据项构成的集合,即 $X \subseteq I$. 设 T 为全体事务集,事务数据库中每条事务 t_i 都有 $t_i \in T$ 且 $t_i \subseteq I$. 一条事务包含项集 X , 当且仅当 $X \subseteq t_i$. 项集 X 的支持数,记作 $\text{sup}(X)$, 表示事务集 T 中包含 X 的事务数. X 在事务集合 T 中的支持度为 X 的支持数 $\text{sup}(X)$ 与 T 中事务总数 $\text{count}(T)$ 的比值. 关联规则是指数据项集间存在的规则,形式为 $X \rightarrow Y$, 其中 $X \subseteq I, Y \subseteq I$ 且 $X \cap Y = \Phi$, X 称为规则前项, Y 称为规则后项, X, Y 中的项可以是本体中不同层次上的概念. 规则 $X \rightarrow Y$ 的支持度定义为事务集合 T 中同时包含项集 X 和 Y 的事务数,记作 $\text{sup}(X \cup Y)$. 关联规则挖掘是在事务中挖掘置信度不小于给定值的规则的知识发现过程. 规则 $X \rightarrow Y$ 的置信度定义为:

$$\text{conf}(X \rightarrow Y) = \frac{\text{sup}(X \cup Y)}{\text{sup}(X)}.$$

2.1 本体概念关系

定义 1 (本体) 本体是概念模型的明确规范和说明^[10]. 本体为五元组^[11], $O(C, R, H, P, A)$. 其中 C 代表全体概念集合; R 代表所有非层次关系集合; H 代表本体层次结构, 记录了概念之间层次信息, $H(c_i, c_j)$ 表示概念 c_i 是概念 c_j 的子节点; P 代表属性集合; A 代表本体公理. 如图 1 所示.

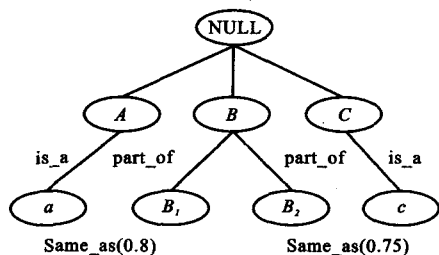


图1 本体概念关系

定义 2 (祖先项集) 设有概念集合 $Y \subseteq X, X \subseteq C$, 如果 Y 中的每一项 y_j, X 中都存在项 x_i , 满足关系 $H(y_j, x_i)$, 即 $Y = \{\forall y_j | (y_j, x_i) \in H, \text{且 } x_i \in X, \text{其中 } j = 0, 1, 2, \dots, m; i = 0, 1, 2, \dots, n\}$, 则称项集 X 是项集 Y 的祖先, 记作 $\text{sub}(Y, X)$. 如果不存在 $Z \subseteq C$, 使得 $\text{sub}(Y, Z)$ 和 $\text{sub}(Z, X)$ 成立, 则称项集 X 是项集 Y 的直接祖先.

定义 3 (is_a 关系) 设 c_i 和 c_j 是 C 中的概念, 如果 c_i 是 c_j 的实例或者 c_i 是 c_j 的子概念, 则认为 c_i 、

c_j 是 is_a 关系, 记作 $\text{is_a}(c_i, c_j)$. 在图 1 中概念 a 、 A 和概念 c 、 C 满足 is_a 关系, 分别表示为 $\text{is_a}(a, A), \text{is_a}(c, C)$. is_a 关系是本体中的层次关系, 即 $\text{is_a}(c_i, c_j) \in H$. is_a 关系是一个偏序关系.

定义 4 (same_as 关系) 设 c_i 和 c_j 是 C 中的概念, 如果 c_i, c_j 的语义相似度不小于给定常数 R , 则认为概念 c_i, c_j 是 same_as 关系, 记作 $\text{same_as}(c_i, c_j)$. 表示两个概念语义相似度足够大, 这时可以认为概念 c_i, c_j 在给定相似度阈值 R 下是同义概念. 图 1 中概念 a, B_1 和概念 c, B_2 满足 same_as 关系, 分别表示为 $\text{same_as}(a, B_1), \text{same_as}(c, B_2)$. same_as 关系是本体中非层次关系, 即 $\text{same_as}(c_i, c_j) \in R$. same_as 关系是一个等价关系.

一个概念与它本身的相似度为 1, 即 $\text{same_as}(c_i, c_i) = 1$. 如果两个概念完全没有相似性, 那么这两个概念的相似度为 0, 即 $\text{same_as}(c_i, c_j) = 0$. 如果有 $\text{same_as}(c_i, c_j), \text{same_as}(c_j, c_k)$, 那么 $\text{same_as}(c_i, c_k) = \text{same_as}(c_i, c_j) \times \text{same_as}(c_j, c_k)$.

定义 5 (part_of 关系) 设 c_i 和 c_j 是 C 中的概念, 如果概念 c_i 是概念 c_j 意义的一部分, 则认为概念 c_i, c_j 是 part_of 关系, 记作 $\text{part_of}(c_i, c_j)$. part_of 关系可以表示概念 c_i 相对于概念 c_j 的重要性. 给定常数 R , 如果 $\text{part_of}(c_i, c_j)$ 不小于 R , 则认为概念 c_i 较之于概念 c_j 重要性足够大, 某些情况下可以代表概念 c_j . 图 1 中概念 B_1, B 和概念 B_2, B 满足 part_of 关系, 分别表示为 $\text{part_of}(B_1, B), \text{part_of}(B_2, B)$. part_of 关系是本体中的层次关系, 即 $\text{part_of}(c_i, c_j) \in H$. part_of 关系是一个偏序关系.

每个概念都是它本身的一部分, 用 $\text{part_of}(c_i, c_i) = 1$ 表示. 两个不相关的概念不可能组成 part_of 关系, $\text{part_of}(c_i, c_j) = 0$ 表示概念 c_i 与概念 c_j 不存在整体与部分的关系. 概念的一部分的一部分是这个概念的一部分, 即如果 $\text{part_of}(c_i, c_j), \text{part_of}(c_j, c_k)$ 成立, 有 $\text{part_of}(c_i, c_k)$ 成立, 且 $\text{part_of}(c_i, c_k) = \text{part_of}(c_i, c_j) \times \text{part_of}(c_j, c_k)$.

2.2 多层关联规则的语义划分

根据语义本体概念间的关系, 文中把关联规则冗余划分三类: 子规则冗余、同义规则冗余、可规约规则冗余.

定义 6 (子规则) 规则 r_1 是规则 r_2 的子规则, 当规则 r_1 中的前件或后件中有项对应与规则 r_2 中前项或后项中某项满足 is_a 关系, 且规则 r_1, r_2 其他的前件和后件的项满足 $\text{same_as}(c_i, c_j) \geq R$, 其

中 R 是最小相似度权值.

在冗余处理的时候删掉子规则,保留父规则.

定义 7 (同义规则) 规则 r_1 与规则 r_2 是同义规则,当且仅当规则 r_1 的前件或后件中的项 c_i 与规则 r_2 的前件或后件中的项 c_j 满足 $\text{same_as}(c_i, c_j) \geq R$, 其中 R 是最小相似度权值.

冗余处理的时候,把规则集合中表述相同意义的规则都分别归类,在输出结果的时候,选择其中的一条作为输出,并注明其它同义规则以及规则相似度.规则相似度取规则中各同义项语义相似度的最小值,即 $\text{same_as}(r_1, r_2) = \min\{\text{same_as}(c_{i1}, c_{j1}), \text{same_as}(c_{i2}, c_{j2}), \dots\}$.

定义 8 (部分规则) 规则 r_1 是规则 r_2 的部分规则,当且仅当 r_1 的前项或后项中有项 c_i 与规则 r_2 中的前项或后项中某项 c_j 满足 $\text{part_of}(c_i, c_j) \geq R$, 其中 R 是最小比重值权值,且规则其它的前件和后件的项 c_g, c_k 满足 $\text{same_as}(c_g, c_k) \geq R'$, 其中 R' 是最小相似度权值.

如果规则 r_1 是规则 r_2 的部分规则,规则 r_1 可规约为规则 r_2 ,则认为规则 r_1 是冗余规则,仅保留规则 r_2 ,使关联规则挖掘结果更加具有概括性.

3 基于语义划分的多层次关联规则冗余处理

关联规则挖掘会产生大量的规则,有时候甚至多达数十万条,要想从如此巨大的规则集合中结合语义信息搜索冗余规则无疑需要很大的运算量,为能更快速准确地对规则进行冗余处理,文中提出前缀树扫描方法来减少冗余处理过程的运算复杂度,提高处理结果的准确度.

方法主要分为 3 部分:(1)按照规则前项对规则进行预处理,用规则前项中的项为节点构建前缀树.这步完成后,把所有规则都压缩到规则前缀树集合中.(2)结合语义本体遍历每棵前缀树.从前缀树根节点到的其它节点的所有路径都有可能是一条规则的前项,从根节点开始遍历前缀树的每个节点同时查找本体,找出这个节点的关联节点,组成项目列表,如果该项目列表能构成规则前项,则把它加入冗余规则候选集合中.这步完成后每条规则都被加入相应的冗余规则候选集合.(3)扫描每个冗余规则候选集合,进行相应的冗余处理.

3.1 构建前缀树

createPreTree (ruleset)

输入:规则集合 ruleset

输出:包含所有规则的前缀树集合 prefixTrees

get all premise from ruleset;

Set < Set < premise > > sortedPre = sortAndGroup (premise);

for preSet in sortedPre

get the first item ITEM

root = buildTree (ITEM);

current = root;

for premise in preSet

if premise not equal current

if current. childs not contain premise

current. childs = $\cup$ {premise};

End if

current = premise

End if

End for

prefixTrees = $\cup$ {root};

End for

使用规则前项中的项为节点构造规则前缀树.

首先,对每条规则的前项进行排序预处理,把第一个项目相同的规则放在一个集合中;对于每个集合,用集合中每条规则前项都含有的相同第一个项目作为前缀树的根节点,依次扫描集合中的每条规则前项,构造前缀树,使得前缀树从根节点到树中其它节点的路径都对应着规则前项.这步需要遍历规则集合中所有的规则,完成后所有的规则都被包含在前缀树集合中.

3.2 结合语义本体遍历前缀树

scanPrefixTree (pfnode, ontology, relType, weight, premises)

输入:前缀树节点 pfnode,本体 ontology,关联类型 relType,weight 关联权值,premises 存储的前缀树从根节点到其他节点的路径集合,表示已经发现的所有规则前项

输出:冗余规则候选集合 candreds

for path in premises

path = $\cup$ {pfnode};

if path is premise of a rule

candred = $\cup$ {path};

paths = $\cup$ {path};

End if

relNodes = ontology. find (pfnode, relType, weight);

if relNodes is not null

for renode in relNodes

newPath ← path. deepclone();

newPath. pop();

newPath = $\cup$ {renode};

if newPath is premise of a rule

```

    candred =  $\cup \{newPath\}$ ;
    paths =  $\cup \{newPath\}$ ;
  End if
End for
End if
candreds =  $\cup \{candred\}$ ;
End for
childs ← fnode.getchilds();
if child is not null
  for child in childs
    scanPrefixTree(child, ontology, relType, paths);
  End for
End if

```

结合语义本体遍历规则前缀树,这步是冗余处理的核心.输入前缀树根节点 root,本体 ontology,关联类型 relType,关联权值 weight,采用深度优先方式遍历.从前缀树根节点开始,记录从根节点到其它节点路径上的所有节点,如果这些节点能够成一条规则前项,则建立冗余候选集合.同时遍历本体,返回与当前节点有指定关联类型 relType 且权值大于 weight 的关联节点,如果关联节点能够与当前节点对应的路径上除当前节点的所有节点构成规则前项,则把这条规则加入到相应的冗余候选集合,递归地遍历当前节点的每个子节点.这个过程总的时间代价大约为 $O(n^2)$.如此,遍历完前缀树后,所有的规则都被加入到相应的冗余候选集合中.

3.3 对冗余规则候选集合进行冗余处理

```

reduceRedundancy (candred)
输入:冗余规则候选集合 candred
输出:无冗余规则集合 rules
for rule in candred
  get consequent of rule
  get relation ITEMS of consequent
  if conSet is not contain consequent and ITEMS
    conSet =  $\cup \{con\}$ 
    rules =  $\cup rule$ 
  End if
End for

```

对每个冗余候选集合进行处理.冗余候选集合中每条规则的前项都是具有前面定义的某一类关系的项集,这时只需要考察候选中每条规则的后项.如果某两个候选的后项也符合这类关系,那么这条规则被认为是这类型的冗余规则,进行相应的处理.

最后用某手机订阅服务数据集作为实验数据进行试验.实验表明,使用本文提出的冗余处理方法能

有效消除多层关联规则冗余,使得挖掘出的规则更加符合实际情况,在实际应用更加具有指导意义;同时通过处理冗余和不处理冗余挖掘时间耗费的对比,表明文中提出的方法在时间耗费上也是可以接受的.

4 结束语

文中针对多层次关联规则挖掘产生的冗余,提出基于语义划分的冗余处理方法,根据语义描述信息将关联规则冗余划分为3类:子规则冗余、同义规则冗余、可规约规则冗余.进行冗余处理的时候删掉冗余的子规则,对同义规则进行归类视为具有相同意义的规则,把只表述了部分信息的可归约规则规约到更高层次.

文中提出的基于语义划分的多层关联规则冗余处理方法对冗余的处理完全基于语义本体概念间的关系,下一步工作是在多层关联规则冗余处理过程中引入规则置信度,对冗余候选集合中规则进行处理的时候,把规则的置信度作为判断标准之一,处理置信度较低的规则,保留置信度较高的规则.

参考文献:

- [1] Yue Xu, Yuefeng Li, Gavin Shaw. Concise representations for approximate association rules[J]. IEEE International Conference on Systems, Man and Cybernetics, 2008(2):94 - 101.
- [2] Abdelaziz Berrado · George C Runger. Using metarules to organize and group discovered association rules[J]. Data mining and knowledge discovery, 2007(14):409 - 431.
- [3] Gavin Shaw, Yue Xu, Shlomo Geva. Utilizing non-redundant association rules from multi-level datasets[C]// IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. Sydney, 2008: 681 - 684.
- [4] Claudia Marinica, Fabrice Guillet, Henri Briand. Post-processing of discovered association rules using ontologies[J]. IEEE International Conference on Data Mining Workshops, 2008(87):126 - 133.
- [5] Ping Chen, Rakesh Verma, Janet C Meininger, et al. Semantic analysis of association rules[C]// Association for the Advancement of Artificial Intelligence. USA, 2008.
- [6] Rafael Garcia Miani, Cristiane A Yaguinuma, Marilde T P Santos, et al. NARFO algorithm mining non-redundant and generalized association rules based on fuzzy ontology[C]// ICEIS. Italy:Milan, 2009:415 - 426.

实验结果表明:GPU 的计算时间相对于数据从 host 端向 device 段的转移和将结果回传到 host 端的时间,非常小,即 GPU 资源在大部分时间处于空闲状态.应该充分发掘 GPU 的使用率,这有待于进一步研究.

7 结束语

视频图像中块效应评价算法具有密集的计算任务和工作量大的特点,文中通过对相同文件的视频图像块效应串行算法和并行算法运行时间以及正确性的比较,结果表明在 GTX280 上实现的并行算法较 CPU 上的串行算法速度提高了近 30 倍.说明利用 CUDA 技术达到了预期的效果,证明了 CUDA 技术在数字图像处理应用中具有极大的潜力,尤其是计算密集型任务处理.但 GPU 资源的充分使用还有待于进一步的发掘,值得下一步探讨研究.

参考文献:

- [1] Kim T K, Paik J K. Fast image restoration for reducing block artifacts based on adaptive constrained optimization [J]. *Journal of Visual Communication and Image Representation*, 1998, 9(3): 234 - 242.
- [2] Wang Z, Bovik A C. *Modern image quality assessment* [M]. USA, Morgan & Claypool Press, San Rafael, 2006.
- [3] Wu H R, Yuen M. A generalized block - edge impairment metric for video coding [J]. *Signal Processing Letters*, IEEE, 1997, 4(11): 317 - 320.
- [4] Minami S, Zakhor A. An optimization approach for removing blocking effects in transform coding [J]. *Circuits and Systems for Video Technology*, IEEE, 1995, 5(2): 74 - 82.
- [5] Triantafyllidis G A, Tzovaras D. Blocking artifact detection and reduction in compressed data [J]. *Circuits and Systems for Video Technology*, IEEE, 2002, 12(10): 877 - 890.
- [6] ZhaoLin Z, Haoshan S. A novel blind measurement of blocking artifacts for H.264/AVC Video [C]// *Image and Graphics*, Fifth International Conference. Japan, 2009.
- [7] 李政芳,周志恒,曾德炉,等.图像块效应评价标准 [J]. *技术探讨*, 2007(5): 59 - 61
- [8] NVIDIA Corporation. NVIDIA CUDA compute unified device architecture programming guid 2.0 [EB/OL]. (2008 - 06 - 07) [2009 - 12 - 11]. http://developer.download.nvidia.com/compute/cuda/2_0/docs/NVIDIA_CUDA_Programming_Guide_2.0.pdf.
- [9] Ying Zhang, Xuejun Yang, Guibin Wang, et al. Scientific computing applications on a stream processor [C]// *Proceedings of IEEE ISPASS*. Austin, Texas, 2008: 105 - 114.
- [10] Owens J D, Houston M, Luebke D, et al. GPU computing [J]. *IEEE*, 2008, 96(5): 879 - 899.

作者简介:

魏晓娟 女, (1985 -), 硕士研究生. 研究方向为并行计算.
陈庆奎 男, (1966 -), 教授, 博士生导师. 研究方向为网络计算、并行理论.
王海峰 男, (1976 -), 博士. 研究方向为并行计算.

(上接第 255 页)

- [7] Zahra Farzanyar, Mohamnadreza Kangavari, Sattar Hashemi. A new algorithm for mining fuzzy association rules in the large databases based on ontology [C]// *Sixth IEEE International Conference on Data Mining - Workshops*, 0 - 7695 - 2702 - 7/06. New York, 2006.
- [8] Claudia Marinica, Fabrice Guillet. Improving post - mining of association rules with ontologies [C]// *The XIII International Conference, Applied Stochastic Models and Data Analysis*. Canada, 2009.
- [9] Joanna Jozefowska, Agnieszka Lawrynowicz, Tomasz Lukaszewski. On reducing redundancy in mining relational association rules from the semantic web [C]// *International Conference on Web Reasoning and Rules Systems*. Germany: karlsruhe, 2008: 205 - 213.
- [10] Gruber T R. A translation approach to portable ontology specifications [J]. *Knowledge Acquisition*, 1993(5): 199 - 220.
- [11] Maedche A. *Ontology learning for the semantic web* [M]. Boston: Kluwer Academic Publishers, 2002.

作者简介:

李帆 男, (1986 -), 硕士研究生. 研究方向为知识发现与数据挖掘.