文章编号:1009-3087(2013)04-0131-09

基于主观兴趣度的关联规则优化算法

牛新征[1],杨 健[2],周明天[1]
(1. 电子科技大学 计算机科学与工程学院,四川 成都 611731;2. 电子科技大学 信息与软件学院,四川 成都 611731)

摘 要:基于兴趣度的规则优化算法通过整合用户领域知识,对规则进行了精简和优化,有效地帮助了用户发现其最感兴趣的规则。但算法仍存在兴趣度计算方式欠妥、用户含义表达受限等问题。提出对兴趣度计算方法的改进,对单模板情况下的计算进行分类讨论,解决了兴趣度计算不合理的问题。同时,新算法引入复合模板的技术,支持对规则的多维分析,丰富了用户含义的表达。通过2组对比实验验证,改进后的基于主观兴趣度的规则优化算法能导出更加合理的兴趣度排序结果,给用户提供更有价值的规则参考。

关键词:关联规则;规则优化;模板;兴趣度
中图分类号:TP393 **文献标志码**:A

Algorithm of Association Rules Optimization Based on the Subjective Interestingness

NIU Xin-zheng[1], *YANG Jian*[2], *ZHOU Ming-tian*[1]
(1. School of Computer Sci. and Eng., Univ. of Electronic Sci. and Technol. of China, Chengdu 611731, China;
2. School of Info. and Software Eng., Univ. of Electronic Sci. and Technol. of China, Chengdu 611731, China)

Abstract: In order to resolve the improper calculation method, an improvement on calculating interestingness was proposed by considering classified situations. Multiple templates were induced, which support multi-dimension analysis and enrich user's implications. With two groups of contrast experiments, the new algorithm produced more reasonable sorting result of interestingness, and supplied users with more valuable rules.

Key words: association rules; optimization; template matching; interestingness

关联规则挖掘作为数据挖掘领域一种强有力的分析工具,能够发现大量数据中项集之间有趣或者相关的联系。但随着海量数据的不断收集与存储,对这些数据进行关联挖掘导出的规则数量也不断增加,这就给分析、决策人员的判断带来了困难。此外,仅基于支持度-置信度框架的传统关联规则挖掘算法(例如 Apriori 算法)并不能指出用户真正感兴趣的规则,给用户对所导出规则的分析带来不便。

基于兴趣度的规则优化方法[1]能有效解决上述问题,帮助用户发现其感兴趣的规则。该方法结合用户领域知识,计算每条规则的兴趣度并做出排序供用户参考。但该种方法存在兴趣度计算方式欠妥、用户含义表达受限等问题。作者对主观兴趣度规则优化算法提出2点改进。改进后的算法不仅解决了一致度计算不合理的问题,还支持用户指定复合模板进行规则分析,丰富了用户含义的表达。最后,通过改进后算法与原算法的对比实验,证明了改进后算法的有效性,体现了其在应用中的重要价值。

1 研究背景

正如前文提到的关联规则挖掘所面临的困难,规则优化则成为了提升规则质量、发现有价值规则的有效手段。因此,许多基于客观或主观的关联规则优化方法被学者们纷纷提出。客观优化方法一般从规则的结构、集合性质、统计结果、离差模型等入手进行分析,而主观优化方法一般利用领域知识、模板、兴趣度等主观量度对规则进行分析[2]。主要的关联规则优化方法有:

收稿日期:2012-08-31
基金项目:华为公司创新研究计划资助项目(YJCB201031RE);四川省科技支撑计划资助项目(2012GZ0061)
作者简介:牛新征(1978—),男,副教授,博士. 研究方向:云计算;数据挖掘. E-mail: xinzhengniu@uestc.edu.cn

在主观分析方面,Piatetsky-Shapiro 首先提出了兴趣度问题[3]。Hoschka 和 Klosgen 首次提出模板的概念[4]。离差分析法被提出用来衡量真实结果与期望结果间的距离,而 Piatetsky-Shapiro 和 Matheus 把离差与兴趣度相结合,分析了离差的兴趣度[5-7]。Klemettinen 等在文献[8]中也提出了规则模板的概念,并使用包含模板和限制模板分别过滤有趣规则和非有趣规则;此外,许多类似规则模板的概念也被纷纷提出,用以描述用户所期望的规则。Srikant 和 Agrawal 首次提出了普适关联规则与领域知识本体论的概念[9-10]。一些学者对基于 Query 的发现有趣规则的方法做出了研究。而在客观分析方面,Toivonen 等在文献[11]中提出规则覆盖集的概念,覆盖集是原始规则集经 RuleCover 算法优化删除后的规则集合。

目前,几乎所有规则优化方法的研究都在集中在上述范畴之内。首先分析了 RuleCover 算法[11-12](一种客观性方法),发现其处理后导出的覆盖集确实囊括了相对普适的规则(即概括性相对较强的规则),此算法效果显著。而主观兴趣度方法则从另外一个角度对规则集进行分析,旨在找出用户所感兴趣的规则[13]。此外,模板作为一种用户用来指定领域知识信息的工具,可被用来辅助主观兴趣度的分析。于是考虑到把主观的兴趣度优化算法与客观的 RuleCover 算法结合起来,便可互补其不足而提高优化能力。但对这种结合后的算法进行实验发现,大量有价值的规则都被过滤掉了,结果并不理想。导致结果不理想的原因有以下几点:首先,RuleCover 算法旨在保留普适规则,而主观兴趣度分析算法旨在发现特殊的、有趣的规则,RuleCover 算法会删除大量特殊规则;其次,覆盖集中规则数量偏少,而主观兴趣度算法在规则数量偏多时效果才显著且更有意义。因此,一个庞大的规则集经过 RuleCover 算法处理后导出的覆盖集中所含规则数量相当有限[12],若再进行主观兴趣度分析则意义不大。

对兴趣度优化算法进行研究更有意义,因为在研究背景上 RuleCover 算法相对更成熟,而兴趣度优化算法还有很大的改进空间。且从用户需求分析,对兴趣度进行研究有以下 2 点意义:1)面对挖掘出来的规则,用户唯一的目标就是去寻找那些特殊的、没有被发现的规则。若仅给出一堆杂乱无序的规则,用户便需要花费较多的时间来分析和发现有价值的规则。2)当规则数量成千上万时,用户希望能快速切入主题、发现价值,而不是面对规则无从下手。因此,主要针对兴趣度优化领域,提出更加完善的主观兴趣度优化算法。

目前的主观兴趣度优化算法存在兴趣度计算方法欠妥、用户含义表达受限等问题。针对这些问题,提出 2 个改进点:1)完善兴趣度的计算方法,添加单一模板下对兴趣度的特殊处理;2)引入模板权重,实现基于复合模板的兴趣度分析,增强规则优化能力。

所讨论的规则集均为无冗余的规则集(即经过冗余删除算法[14]的处理),而不再经过其他算法的处理,后文实验中采用的也是无冗余数据。

2 问题描述

2.1 模板与知识类型

模板是主观兴趣度算法中使用到的一个重要的工具,是用户表达含义的载体。因为对原算法的改进涉及到了对多个模板的支持,参照文献[8]中对模板的描述,给出模板的定义:

定义 1 形如 $A_1,\cdots,A_K \Rightarrow A_{k+1}$(其中,$A_i$ 可以是一个属性名、类名或者如 $C+$、C^* 之类的表达式,$C+$ 表示一个或多个类 C 的实例,C^* 表示零或多个类 C 的实例)的蕴含式,称为模板。

用户使用模板来指定规则中所期望的前件与后件,如果一个关联规则 $B_1,\cdots,B_h \Rightarrow B_{h+1}$ 是上述模板的一个实例,则称这个关联规则匹配上述模板[5]。模板通过标识有趣规则和非有趣规则,可帮助用户提取出有价值的规则。

模板作为用户领域知识的载体,可以表达多种多样的含义。为便于模板形式的统一,参照文献[1]中总结的 3 种知识类型,给出基于总体印象知识的模板、基于相对精确知识的模板这 2 种模板的定义如下:

定义 2 用户因项间关系模糊而给出的不确定的知识模板,称为总体印象知识模板,简称 GI 模板,表示为 $gi[S_1,\cdots,S_m]$(其中,S_k 可以是一个属性名、类名或者一条表达式)。

定义 3 用户知晓项间关系且明确关系方向而给出的相对合理的知识模板,称为相对精确知识模板,简称 RPC,表示为 $rpc[S_1,\cdots,S_m \Rightarrow V_1,\cdots,V_g]$(其中,$S_k$ 与 V_k 可以是一个属性名、类名或者一条表达式)。

这 2 种典型模板作为用户领域知识的载体,将由计算机解析,进而对规则进行相关的分析。

2.2 有趣规则及其兴趣度

对原算法的改进还涉及到对兴趣度计算方法的完善,下面描述和兴趣度相关的问题。

对“一个规则是否有趣”这个问题,难以做出准确的判定。但是可以从关联规则的结构出发,结合规则与模板的匹配情况对不同的规则进行分类,以区分不同类型的“有趣规则”。通过对规则的分类,不仅可以化“有趣”这个抽象概念为具体,还使得对兴趣度的计算更加多样化、合理化。

结合模板的概念给出 4 种有趣规则类型的定义,分别是一致规则、后件不可预知规则、前件不可预知规则、不可预知规则:

定义 4　若规则 R_i 的前件与后件均与指定的模板 U_j 相匹配,则称规则 R_i 为一致规则。

定义 5　若规则 R_i 的前件与指定的模板 U_j 匹配,而后件不匹配,则称规则 R_i 为后件不可预知规则。

定义 6　若规则 R_i 的后件与指定的模板 U_j 匹配,而前件不匹配,则称规则 R_i 为前件不可预知规则。

定义 7　若规则 R_i 的前件与后件均与指定的模板 U_j 不匹配,则称规则 R_i 为不可以预知规则。

针对上述 4 种不同类型的规则,需要有 4 种不同的量度分别对其有趣程度进行定量描述。相应地,给出 4 种兴趣度的定义如下:

定义 8　一致规则 R_i 的兴趣度表示规则前件、后件与指定模板集 U 匹配的程度,用符号 $conf_i$ 表示,称为规则 R_i 的一致度。

定义 9　后件不可预知规则 R_i 的兴趣度表示规则后件与指定模板集 U 不匹配的程度,用符号 $unexpY_i$ 表示,称为规则 R_i 的后件不可预知度。

定义 10　前件不可预知规则 R_i 的兴趣度表示规则前件与指定模板集 U 不匹配的程度,用符号 $unexpX_i$ 表示,称为规则 R_i 的前件不可预知度。

定义 11　不可预知规则 R_i 的兴趣度表示规则前件、后件与指定模板集 U 不匹配的程度,用符号 $unexp_i$ 表示,称为规则 R_i 的不可预知度。

上述 4 种兴趣度的值均在[0,1]范围内,具体计算方法将在下文进行描述。

3 改进的主观兴趣度算法

下面将针对以下 2 个改进点逐一展开研究:1)一致度 $conf_i$ 的计算缺乏特殊情况下的考虑,当指定的模板仅为 GI 时,规则的 $conf_i$ 值几乎全为零而将进行没有意义的排序;2)原算法只能指定一条 GI 或者 RPC 模板进行兴趣度分析,使得用户的表达范围有限。

为便于描述,预先给出相关符号的说明:设有原始关联规则集 $R_0 = \{X_i \Rightarrow Y \mid i = 1,\cdots,n\}$,其中,$X_i \Rightarrow Y$ 为关联规则,R_i 为 R_0 中的一条规则。XN_i 为规则 R_i 中前件 X_i 所含元素个数,YN_i 为规则 R_i 中后件 Y 所含元素个数。设 $GI = \{GI_j \mid j = 0,\cdots,l\}$ 为用户指定的 GI 模板集合,$RPC = \{RPC_j \mid j = 0,\cdots,k\}$ 为用户指定的 RPC 模板集合,$U = \{U_j \mid U_j \subset GI$,或 $U_j \subset RPC, j = 1,\cdots,n\}$ 为指定模板总集。

3.1 复合模板与权重

为支持用户指定多个模板同时进行分析,算法将引入模板权重 $weight$,通过叠加多个模板的带权兴趣度得到最终兴趣度,从而实现多模板的最终兴趣度排序。

为简化算法描述,这里默认模板权重值为 $weight = 1/n$,n 为 U 中模板数。设 X_{ij}、Y_{ij} 分别为 R_i 中前件、后件与 GI_j 或 RPC_j 中前件、后件不匹配程度的度量(作为过渡值而无特殊含义)。TX_i 为 X_{ij} 的权重累计值,TY_i 为 Y_{ij} 的权重累计值。TX_i、TY_i 按如下公式计算:

$$TX_i = TX_i + 1/n \times X_{ij} \tag{1}$$

$$TY_i = TY_i + 1/n \times Y_{ij} \tag{2}$$

而对 X_{ij}、Y_{ij} 的计算,分 GI 模板与 RPC 模板 2 种情况。

1) GI 模板情况下:设 TN_j 为 GI_j 中元素总数(含有 * 扩展标记的元素不计),XM_{ij}、YM_{ij} 分别为 R_i 中前件、后件与 GI_j 中元素相匹配的个数,TM_{ij} 为 GI_j 中已被 R_i 中元素所匹配的元素总数(含有 * 扩展标记的元素不计)。

若 $TN_j = 0$,则 $TM_{ij}/TN_j = 1$,

$$X_{ij} = \begin{cases} \min(XM_{ij}/XN_i, TM_{ij}/TN_j), \\ \quad \text{if } XM_{ij}/XN_i > YM_{ij}/YN_i; \\ XM_{ij}/XN_i, \text{ else } XM_{ij}/XN_i \le YM_{ij}/YN_i \end{cases} \tag{3}$$

$$Y_{ij} = \begin{cases} YM_{ij}/YN_i, \text{ if } XM_{ij}/XN_i > YM_{ij}/YN_i; \\ \min(YM_{ij}/YN_i, TM_{ij}/TN_j), \\ \quad \text{else } XM_{ij}/XN_i \le YM_{ij}/YN_i \end{cases} \tag{4}$$

2)RPC 模板情况下:设 TXN_j、TYN_j 分别为 RPC_j 中前件、后件所含元素总数(含有 * 扩展标记的元素不计),XM_{ij}、YM_{ij} 分别为 R_i 中前件、后件与 RPC_j 中前件、后件所含元素相匹配的个数,TXM_{ij}、TYM_{ij} 分别为 RPC_j 中前件、后件已被 R_i 中前件、后件所匹配的元素总数(含有 * 扩展标记的元素不计)。

若 $TXN_j = 0$,则 $TXM_{ij}/TXN_j = 1$,

若 $TYN_j = 0$,则 $TYM_{ij}/TYN_j = 1$,

$$X_{ij} = \min(XM_{ij}/XN_i, TXM_{ij}/TXN_j) \quad (5)$$

$$Y_{ij} = \min(YM_{ij}/YN_i, TYM_{ij}/TYN_j) \quad (6)$$

可见,*weight* 的引入均匀地综合了多个模板的相关参数,使得能进行下一步的兴趣度计算。

3.2 兴趣度的计算

兴趣度作为对规则有趣程度的度量,从侧面反映了规则对用户而言价值的高低。计算兴趣度的算法增加了仅指定一个 GI 模板情况下对一致度 $conf_i$ 的分类处理。下面一起给出 $conf_i$、$unexpY_i$、$unexpX_i$ 与 $unexp_i$ 的计算方法。

当指定模板仅为 GI 模板时(此时 $i = 1$),计算 $conf_i$ 的公式如下:

$$conf_1 = \begin{cases} 0, & TX_1 = 0, TY_1 = 0; \\ TY_1, & TX_1 = 0, TY_1 \neq 0; \\ TX_1, & TX_1 \neq 0, TY_1 = 0; \\ TX_1 \times TY_1, & \text{others} \end{cases} \quad (7)$$

除此之外,$conf_i$ 仅由下列公式计算:

$$conf_i = TX_i \times TY_i \quad (8)$$

下列公式用于计算 $unexpY_i$、$unexpX_i$、$unexp_i$:

$$unexpY_i = \begin{cases} TX_i - TY_i, & TX_i - TY_i > 0; \\ 0, & TX_i - TY_i \leq 0 \end{cases} \quad (9)$$

$$unexpX_i = \begin{cases} TY_i - TX_i, & TY_i - TX_i > 0; \\ 0, & TY_i - TX_i \leq 0 \end{cases} \quad (10)$$

$unexp_i = 1 - \max(conf_i, unexpY_i, unexpX_i)$。

可见对 $conf_i$ 的计算分了 2 种情况来讨论。若指定模板仅为 GI 模板时仍然使用式(8)计算,会导致 $conf_i$ 的值几乎全为零而进行无意义地排序。

3.3 伪码描述

设含不同类型兴趣度的规则集分别为 R_{conf}、R_{unexpY}、R_{unexpX}、R_{unexp}。结合上面 2 个部分的描述,综合给出以下伪码描述(主要由 3 部分组成):

Algorithm Improved Subjective Interestingness

输入:$R_0, U, weight$

输出:$R_{\text{conf}}, R_{\text{unexpY}}, R_{\text{unexpX}}, R_{\text{unexp}}$

//① 解析 U,根据 U 对 R_0 进行扫描并计数

for $j = 1, \cdots, n$ do

  for $i = 1, \cdots, m$ do

    // 规则 R_i 涉及的参数计数

    count XN_i, YN_i;

    // 根据解析后的模板对规则计数

    if $U_j \in GI$ then

      //*GI* 模板涉及的参数计数

      count $TN_j, XM_{ij}, YM_{ij}, TM_{ij}$;

      // 计算 X_{ij}, Y_{ij} 的值,如果 $TN_j = 0$,则有 $TM_{ij}/TN_j = 1$

      if $XM_{ij}/XN_i > YM_{ij}/YN_i$ then do

        $X_{ij} = \min(XM_{ij}/XN_i, TM_{ij}/TN_j)$;

        $Y_{ij} = YM_{ij}/YN_i$;

      else

        $Y_{ij} = \min(YM_{ij}/YN_i, TM_{ij}/TN_j)$;

        $X_{ij} = XM_{ij}/XN_i$;

    if $U_j \in RPC$ then

      //*RPC* 模板涉及的参数计数

      count $TXN_j, TYN_j, XM_{ij}, YM_{ij}, TXM_{ij}, TYM_{ij}$;

      // 如果 $TXN_j = 0$,则有 $TXM_{ij}/TXN_j = 1$

      $X_{ij} = \min(XM_{ij}/XN_i, TXM_{ij}/TXN_j)$;

      // 如果 $TYN_j = 0$,则有 $TYM_{ij}/TYN_j = 1$

      $Y_{ij} = \min(YM_{ij}/YN_i, TYM_{ij}/TYN_j)$;

//② 加权值 TX_i, TY_i 的计算与各兴趣度的计算

$TX_i = 0; TY_i = 0$; // 赋初值

for $i = 1, \cdots, m$ do

  for $j = 1, \cdots, n$ do

    // 计算 TX_i,默认权重 *weight* 为 $1/n$

    $TX_i = TX_i + 1/n \times X_{ij}$;

    // 计算 TY_i,默认权重 *weight* 为 $1/n$

    $TY_i = TY_i + 1/n \times Y_{ij}$;

  $conf_i = TX_i \times TY_i$; // 计算 $conf_i$

  if $|U| = 1$ and $U_1 \in GI$ do

    if $TX_1 = 0$ and $TY_1 = 0$ do nothing;

    else if $TX_1 = 0$ do $conf_1 = TY_1$;

    else if $TY_1 = 0$ do $conf_1 = TX_1$;

  // 计算 $unexpY_i$,若 $TX_i - TY_i \leq 0$,则 $unexpY_i = 0$

  $unexpY_i = TX_i - TY_i$;

  // 计算 $unexpX_i$,若 $TY_i - TX_i \leq 0$,则 $unexpX_i = 0$

  $unexpX_i = TY_i - TX_i$;

  // 计算 $unexp_i$

  $unexp_i = 1 - \max(conf_i, unexpY_i, unexpX_i)$;

//③ 对含不同类型兴趣度的规则集排序

sort(R_{conf}); sort(R_{unexpY}); sort(R_{unexpX}); sort(R_{unexp});

return $R_{\text{conf}}, R_{\text{unexpY}}, R_{\text{unexpX}}, R_{\text{unexp}}$;

简要介绍上述算法的核心思想:对每条解析后的模板,均扫描一次规则集;扫描过程中,根据模板对每条规则进行多方面的统计;综合规则的各统计值,计算其一致度、后件不可预知度、前件不可预知度、不可预知度;最后,规则集基于其中一种兴趣度的值降序排列,并返回结果。

将通过具体实验,来验证改进后算法的有效性。

4　实验及其结果分析

在上述改进算法的基础上进行编程仿真实验。实验使用 Java 语言,实验平台为 Eclipse3.6。根据对算法做出的 2 点改进,现确立仿真目标如下:通过实验对比原始与改进后的主观兴趣度分析算法处理后的结果,以证明改进后的算法在以下 2 点优于原始算法:1)当仅指定一条 GI 模板时,一致度的值不会出现全为零而进行无意义排序的情况,增强了算法的有效性;2)可以指定多条模板,扩大用户的分析范围,进而增强分析能力。

4.1　数据来源与处理

实验采用数据为 Mushroom Database[15]。蘑菇数据库内含 8 124 条记录,属性项一共有 23 个,具体包括可食用性、菇帽形状、菇帽表面、菇帽颜色等等共 23 种。

实验使用的关联规则挖掘工具为 Weka 3.6。原始的 Mushroom Database 数据在预处理成 csv 格式文件之后,被导入到 Weka 3.6 的 Explorer 功能模块中进行 Apriori 关联规则挖掘。挖掘的具体参数设置见表 1。

表 1　Weka 3.6 关联规则挖掘参数设置

Tab.1　Preferences of mining association rules

属性名	属性值
Associator	Apriori
car	false
classIndex	-1
delta	0.05
lowerBoundMinSupport	0.1
metricType	Confidence
minMetric	0.9
numRule	10 000
outPutItemSets	false
removeAllMissingCols	false
significanceLevel	-1.0
upperBoundMinSupport	1.0
verbose	false

可见,挖掘过程采用无冗余模式,导出的规则是无冗余的。

挖掘结束后,从 Weka3.6 导出了按置信度(Confidence)排序的 10 000 条规则。对这 10 000 进行了简单的预处理,即去除了无关数据而仅保留下这 10 000 条规则。下面的实验都将基于这 10 000 条规则。部分规则见表 2(前 3 条与后 3 条规则)。

表 2　Weka3.6 导出关联规则示例

Tab.2　Examples of association rules exported

规则编号	规则前件	前件支持计数	规则后件	后件支持计数	置信度
1	veil - color = white4	7 924	veil - type = partial	7 924	1
2	gill - attachment = free	7 914	veil - type = partial	7 914	1
⋮	⋮	⋮	⋮	⋮	⋮
3	gill - attachment = free veil - color = white4	7 906	veil - type = partial	7 906	1
9998	gill - attachment = free gill - spacing = close gill - size = broad stalk - surface - below - ring = smooth2 veil - color = white4	2 680	stalk - surface - above - ring = smooth1	2 608	0.97
9999	gill - spacing = close gill - size = broad stalk - surface - below - ring = smooth2 veil - color = white4	2 680	gill - attachment = free stalk - surface - above - ring = smooth1	2 608	0.97
10000	gill - attachment = free gill - spacing = close gill - size = broad stalk - surface - below - ring = smooth2	2 680	stalk - surface - above - ring = smooth1 veil - color = white4	2 608	0.97

表2简要说明:规则编号为按置信度值进行排序的序号,置信度取值范围为[0,1];前、后件支持计数为其前、后件所含项在原始事务集中出现次数,在本次实验中没有意义,将被忽略;“=”前部为属性类别,后部位其属性值。

4.2 模板格式与示例

实验中采用的模板格式主要由3部分组成:1)“gi/GI”或“rpc/RPC”标识符:用于标明模板类型;2)中括号“[]”:“[]”内字符为指定模板信息的主体;3)信息类型标识符“@”、“#”、“*”、“+”、“==>”、“,”和大括号“{}”等:“@”标志指定信息为属性值,“#”标志指定信息为属性类,“*”标志指定信息出现0次或1次,“+”标志指定信息出现1次或多次,“==>”标志前件与后件的分隔,“,”用于分隔各个指定项,“{}”将2个及以上指定项组合成整体。给出2个示例说明实验中采用的模板格式:

示例1:

gi[veil-color=white@,ring-number#,bruises#+,{ring-number=one@,veil-type=partial@}*]。

说明:指定gi模板,要求规则中包括:1个项为veil-color且其值为white,1个项为ring-number,1个或多个项为bruises,0个或1个项组合为ring-number=one且veil-type=partial。

示例2:

RPC[veil-color# ==> veil-type#]。

说明:指定rpc模板,要求规则中包括:前件中1个项为veil-color,后件中一个项为veil-type。

4.3 实验结果

进行了2组实验:第1组为单GI模板的对比实验,第2组为多模板对比实验。

第1组实验指定单GI模板,将原算法与改进后算法导出的结果进行对比,以验证仅指定一条GI模板时,改进后算法的一致度 $conf_i$ 的值不会出现全为零而进行无意义排序的情况。这里采用的原始算法仅按式(8)计算 $conf_i$ 的值,而不再使用式(7)计算。

第2组实验中,改进后算法指定复合模板进行实验,而原算法指定单模板进行实验。通过对这2种算法导出结果的比较,来验证改进后的算法通过指定多模板可以使用户表达更加丰富的含义,进而增强分析能力、提升结果的有效性。

1)实验1。指定模板1为:

模板1:

gi[stalk-color-above-ring=white2@,veil-type=partial@]。

含义说明如下:模板要求规则中出现项stalk-color-above-ring=white2和项veil-type=partial,即研究菌环以上菌柄颜色为白色、菌衣类型为局部这2种属性值之间的关系。

由于只需验证按一致度 $conf_i$ 排序的结果,将略去其他3种兴趣度的排序结果。为便于比较,以表格形式给出了原始算法与改进后的算法导出的结果(由于规则繁多,均截取了排名前3的规则)。

表3 改进后算法按一致度 $conf_i$ 排序结果

Tab.3 Ranking of association rules by $conf_i$ from improved algorithm

$conf_i$ 排名	编号	规则前件	规则后件	$conf_i$
1	39	stalk-surface-above-ring=smooth1, veil-color=white4	veil-type=partial	0.50
2	41	stalk-surface-above-ring=smooth1, veil-color=white4, gill-attachment=free	veil-type=partial	0.50
3	43	stalk-surface-above-ring=smooth1, veil-color=white4	veil-type=partial, gill-attachment=free	0.50

表4 原算法按一致度 $conf_i$ 排序结果

Tab.4 Ranking of association rules by $conf_i$ from original algorithm

$conf_i$ 排名	编号	规则前件	规则后件	$conf_i$
1	181	stalk-color-below-ring=white3, veil-type=partial, gill-attachment=free	veil-color=white4	0
2	182	stalk-color-below-ring=white3, veil-color=white4	veil-type=partial, gill-attachment=free	0
3	183	stalk-color-below-ring=white3, veil-type=partial	veil-color=white4, gill-attachment=free	0

结果分析:从表 4 中很容易看出,所有规则的一致度均为 0,这意味着所有规则均与模板 1 指定的项不匹配。然而,规则 181 的前件、规则 182 的后件、规则 183 的前件都出现了 veil - type = partial 项,表明这 3 条规则与模板 1 并非完全不匹配。所以原算法的一致度值排序与实际结果相违,是没有意义的排序。再看表 3,发现前 3 条规则的一致度均为 0.50,因为规则 39、41、43 的前后件均有项与模板 1 匹配,是正确而有意义的排序。

2)实验 2。指定模板 2 为复合模板,用于改进后算法的实验:

模板 2:

rpc[classes# == > veil - type#],

rpc[classes# == > veil - color#]。

含义说明:要求规则前件出现项 classes,而规则后件出现项 veil - type 或 veil - color,即研究可食用或有毒蘑菇的菌衣颜色、菌衣类型等特性。

若只能指定 1 个模板,则难以表达上述的含义。若编写模板为 rpc[classes# == > veil - type#,veil - color#],或 rpc[classes# == > veil - type# * ,veil - color# *],虽然表面上这 2 个模板与模板 2 很相似,但它们之间仍有细微的差别。简单地合并多个模板成为单个模板,是表达不出复合模板所能表达的含义的。

指定模板 3 为 1 个 RPC 模板,用于原算法的实验。不妨选取上述 2 个单一模板中的 1 个:

模板 3:

rpc[classes# == > veil - type#,veil - color#]。

为便于比较,以表格形式给出原始算法与改进后的算法导出的结果(由于规则繁多,截取了排名前 3 或前 4 的规则,且选取了效果显著的 $conf_i$ 与 $unexpY_i$ 排序以便于读者观察)。通过对比下面的实验结果,能发现复合模板的确表达了更加丰富的含义,并且其结果优于原始算法单模板导出的结果。

表 5　改进后算法按一致度 $conf_i$ 排序结果

Tab. 5　Ranking of association rules by $conf_i$ from improved algorithm

$conf_i$ 排名	编号	规则前件	规则后件	$conf_i$
1	352	classes = poisonous, gill - spacing = close, gill - attachment = free	veil - color = white4, veil - type = partial	0.17
2	485	ring - number = one, classes = poisonous, veil - type = partial, gill - spacing = close	veil - color = white4, gill - attachment = free	0.13
3	486	ring - number = one, classes = poisonous, gill - spacing = close, gill - attachment = free	veil - color = white4, veil - type = partial	0.13
4	1258	habitat = woods, veil - type = partial	veil - color = white4	0

表 6　原算法按一致度 $conf_i$ 排序结果

Tab. 6　Ranking of association rules by $conf_i$ from original algorithm

$conf_i$ 排名	编号	规则前件	规则后件	$conf_i$
1	320	classes = poisonous, veil - color = white4, gill - attachment = free	veil - type = partial	0.17
2	335	ring - number = one, classes = poisonous, gill - attachment = free	veil - type = partial	0.17
3	339	classes = poisonous, veil - color = white4, gill - spacing = close	veil - type = partial	0.17
4	340	classes = poisonous, veil - type = partial, gill - spacing = close	veil - color = white4	0.17

结果分析:表 5、6 中的规则前件均含有 classes 属性项。表 6 的规则后件中,veil - type 或 veil - color 属性项均有出现,但只有 1 项;而表 5 的规则后件中,属性项的出现更丰富,既有 veil - type 或 veil - color 中的一个出现,又有 2 者的组合出现。所以,改进后的算法能全面地反映模板 2 的含义,而与模板 2 含义相似的单一模板 3 却无法丰富体现其含义。

可见,支持多模板能扩大表达的含义,增强分析优化的能力。

表7 改进后算法按后件不可预知度 $unexpY_i$ 排序结果

Tab. 7 Ranking of association rules by $unexpY_i$ from improved algorithm

$unexpY_i$ 排名	编号	规则前件	规则后件	$unexpY_i$
1	226	classes = edible, veil - color = white4, veil - type = partial	gill - attachment = free	0.33
2	227	classes = edible, veil - color = white4, gill - attachment = free	veil - type = partial	0.33
3	320	classes = poisonous, veil - color = white4, gill - attachment = free	veil - type = partial	0.33

表8 原算法按后件不可预知度 $unexpY_i$ 排序结果

Tab. 8 Ranking of association rules by $unexpY_i$ from original algorithm

$unexpY_i$ 排名	编号	规则前件	规则后件	$unexpY_i$
1	344	ring - number = one, classes = poisonous, veil - color = white4, veil - type = partial	gill - attachment = free	0.25
2	460	classes = edible, gill - size = broad, veil - color = white4, veil - type = partial	gill - attachment = free	0.25
3	468	ring - number = one, classes = poisonous, veil - type = partial, gill - spacing = close	gill - attachment = free	0.25

结果分析:表7、8中,规则前件均含有classes项。表8中,规则后件都不含有模板3中后件所指定的项。而表7中,有2个规则后件含有模板2中后件所指定的项。从主观上分析,用户更希望看到的是表7中的规则,即与模板中指定的项半分相似的规则;而与模板中指定项完全无关的规则(如表8中规则),将失去模板的约束作用。所以,支持复合模板能给用户带来更多所感兴趣的规则,增强了优化能力。

4.4 实验评估

主观兴趣度优化算法由于涉及到较多主观内容,难以从客观方面对结果做出评价,所以上述实验采用了对比实验法,并结合主观认知进行评价。针对上述实验,做出如下评估:

1)当指定一条GI模板时,改进后的算法中 $conf_i$ 值不会出现全为零而进行无意义排序的情况。

2)改进后算法可以指定多条模板,扩大用户的分析范围,进而增强辅助分析能力。

3)考虑到主观因素,例如模板的指定、权重的选取等,改进后的算法导出的结果并不能在客观上保证绝对正确。更重要的是,这种主观分析方法,确实能把一些有趣的规则通过兴趣度排序而出现在最顶端。

综上所述,改进后的算法有效避免了按 $conf_i$ 值无意义排序的情况,且多模板的支持使用户能表达更加丰富的含义,有效增强了算法的优化能力。

5 结 语

由于关联规则挖掘导出的规则存在数量多、质量低等问题,对已经发现的规则进行优化有着重要的研究意义。从兴趣度入手对主观兴趣度优化算法做出进一步完善,从改进兴趣度计算方法和扩大含义表达范围两方面提高了算法的优化能力。通过实验验证,改进后的主观兴趣度分析算法其优化能力得到了有效增强。

今后,会继续研究更加复杂的模板分析方法以扩大其分析范围,并将不断完善兴趣度衡量的方法,使能满足用户的多元需求。此外,还将引入大量其他领域的数据进行测试与评估以增强算法的普适性。

参考文献:

[1] Liu B, Hsu W, Chen S, et al. Analyzing the subjective interestingness of association rules[J]. Intelligent Systems and their Applications, 2000, 15(5): 47 - 55.

[2] He Zhi. Research on the optimization of association rules [D]. Beijing: Beijing Jiaotong University, 2006. [贺志. 关联规则优化方法的研究[D]. 北京:北京交通大学, 2006.]

[3] Piatesky-Shapiro G. Discovery, analysis, and presentation of strong rules[M]//Piatesky-Shapiro G, Frawley W. Knowledge Discovery in Databases. Cambridge: MIT Press, 1991:

229 – 248.

[4] Hoschka P, Klosgen W. A support system for interpreting statistical data [M]//Piatetsky-Shapiro G, Frawley W J. Knowledge Discovery in Database. Cambridge: MIT Press, 1991:120 – 127.

[5] Piatetsky-Shapiro G, Matheus C J. The interestingness of deviations[M]//Fayyad U M, Uthurusamy R. Knowledge Discovery in Databases. Seattle: AAAI Press, 1994:25 – 36.

[6] Han J W, Kamber M. Data mining-concepts and techniques [M]. 范明,孟小峰,译. 北京:机械工业出版社,2001:149 – 184.

[7] Jiang Xin, Li Weihua, Shi Haobin, et al. Distance-based optimization approach for correlation analysis of association rule mining[J]. Computer Engineering and Applications, 2009, 45(7):138 – 140. [蒋欣,李伟华,史豪斌,等. 基于距离的关联规则相关性分析优化方法[J]. 计算机工程与应用,2009,45(7):138 – 140.]

[8] Klemettinen M, Mannila H, Ronkainen, et al. Finding interesting rules from large sets of discovered association rules [C]//Adam N R, Bhargava B K, Yesha Y. CIKM '94 Proceedings of the Third International Conference on Information and Knowledge Management. New York: ACM, 1994: 401 – 407.

[9] Srikant R, Agrawal R. Mining generalized association rules [C]//Dayal U, Gray P M D, Nishio S. Proceedings of the 21st International Conference on Very Large Databases. 1995:407 – 419.

[10] Marinica C, Guillet F, Briand H. Post-processing of discovered association rules using ontologies[C]. ICDMW'08, IEEE International Conference on Data Mining Workshops, 2008:126 – 133.

[11] Toivonen H, Klemettinen M, Ronkainen, et al. Pruning and grouping of discovered association rules[C]//The ECML-95 Workshop on Statics, Machine Learning, and Knowledge Discovery in Databases. Heraklion, 1995:47 – 52.

[12] Cristofor L, Simovici D. Generating an informative cover for association rules[C]//The 2002 IEEE International Conference on Data Mining (ICDM02). 2002:597 – 600.

[13] Zheng Yan, Zhu Qunxiong. Improved FP-TREE algorithm and application based on user interests[J]. Computer Engineering and Applications, 2012, 48(11): 143 – 147. [郑滟,朱群雄. 基于用户兴趣的 FP – TREE 算法的改进及应用[J]. 计算机工程与应用,2012,48(11):143 – 147.]

[14] Wei Suyun, Ji Genlin, Qu Weiguang. Pruning and clustering discovered association rules[J]. Mini-Micro System, 2006, 27(1):110 – 113. [韦素云,吉根林,曲维光. 关联规则的冗余删除与聚类[J]. 小型微型计算机系统,2006,27(1):110 – 113.]

[15] Blake C L, Merz C J. Irvine: Repository of machine learning databases[EB/OL]. [1998]. http://www.ics.uci.edu/~mlearn/MLRepository.html.

(编辑　杨　蓓)