基于频次有效长度的加权关联规则挖掘算法研究*

张　勇　李树青　程永上
(南京财经大学信息工程学院　南京 210046)

摘要: 【**目的**】通过对数据库中项在重要程度上存在的差异性进行分析, 解决传统关联规则挖掘算法挖掘大量冗余无价值规则的问题。【**方法**】在具有时态约束的序列上, 结合频次有效长度方法挖掘非加权关联规则, 引入加权方法, 利用滑动窗口技术在时序序列上挖掘稀有加权关联规则。【**结果**】根据频次有效长度的加权关联规则挖掘算法所挖掘出的加权时序关联规则能够较为准确地进行推荐, 推荐预测的准确度由 62%提升至 69%。【**局限**】由于滑动窗口每次滑动一个单位长度, 加之窗口中生成的规则数量较多, 导致挖掘算法在进行规则挖掘时执行时间较长。【**结论**】本文方法所挖掘出的加权时序关联规则能使推荐精度得到提升, 并为关联规则挖掘方法提供新的研究思路。

关键词: 数据挖掘　关联规则　频次长度　滑动窗口
分类号: G354
DOI: 10.11925/infotech.2096-3467.2018.0999

1　引　言

关联规则是数据挖掘研究领域的一个主要分支, 而加权关联规则更是关联规则挖掘的一个重要课题。算法的任务主要是解决数据间的关联问题和模式的挖掘[1]。不仅如此, 大多数研究者关注点仅仅在于加权关联规则挖掘算法改进方面的研究, 忽视了传统的关联规则挖掘算法面临的两个亟待解决的问题: 一是如 Apriori 算法仅仅考虑数据库中项目出现的频率, 却忽视了数据库中每个项目的重要程度的差异性; 二是如何处理项目权重和衡量权重标准使得挖掘出的关联规则更加贴合实际需要。针对以上两个问题, 研究发现通过设置权值并且允许权重与事务中的每个项目相关联来反映交易中每个项目的兴趣, 并基于加权关联规则挖掘方法开发新的推荐算法以扩展传统关联规则挖掘算法[2]。在引入权值方法进行规则挖掘后, 也只是对传统关联规则挖掘的改进, 若要有所突破则需要创新规则挖掘算法本身, 因此本文提出一种结合频次有效长度挖掘加权关联规则的算法, 挖掘非冗余稀有加权关联规则, 提升推荐准确度。

2　文献综述

目前已有许多方法改进挖掘频繁项目集的效率以及通过加权方法进行强关联规则的挖掘, 但是传统模型挖掘关联规则数量通常非常大, 很难挖掘非冗余关联规则[3]。Khan 等[1]提出 Weighted Utility Association Rule Mining (WUARM)框架, 能够以混合方式处理项目权重和效用(Utility)。此框架可以集成到挖掘过程中, 这与大多数效用和加权关联规则挖掘算法不同, 克服了一起使用权重和效用时面临的挑战, 特别是向下闭包属性的无效。Zhai 等[3]提出一种从加权交易数据库中挖掘加权关联规则的有效方法, 利用不可分辨性矩

通讯作者: 李树青, ORCID: 0000-0001-9814-5766, E-mail: leeshuqing@163.com。
*本文系国家社会科学基金项目“基于大数据分析的数字图书馆个性化服务模式创新研究”(项目编号: 16BTQ030)的研究成果之一。

阵快速找到格子(Lattice)的所有节点; 同时提出一种基于矩阵结果构建频繁权重闭项集格(Frequent Weighted Closed Itemsets Lattice, FWCIL)的增量算法。Ouyang[4]提出一种利用滑动窗口在数据流上挖掘稀有加权关联规则的算法, 并且利用滑动窗口在在线实时数据流中挖掘加权关联规则。李成军等[5]提出一种新的计算加权关联规则挖掘中支持度和置信度的方法, 该算法通过加权体现各项目的实际重要性, 同时能保持 Apriori 向频繁项向下封闭的特性。

虽然国内外学者对于加权关联规则挖掘算法已经取得了很好的成果, 但是在进行加权关联规则挖掘时却忽视了对具有时态约束的长事务序列的思考; 在挖掘方法上受到传统挖掘方法的约束, 缺少对数据库中数据本身的考量与分析, 如数据中涉及周期性、泛化性和多层次, 此时需采用不同的挖掘方法以及合适的加权方法。

本文提出一种结合频次有效长度挖掘加权关联规则的算法。

(1) 实验探索发现时序序列中存在频次高低之分的类型项, 并发现该频次最高的类型项每次出现间隔的节点数存在一个最合适的长度, 于是可以在这个长度内挖掘出关联规则。

(2) 设置加权方法, 不断去除冗余关联规则。

(3) 确定使准确度最高的加权支持度和加权置信度阈值, 并形成不同区间, 从而挖掘出高价值的稀有关联规则进行推荐实验。

3 相关研究工作

3.1 关联规则与加权关联规则概述

(1) 关联规则

设 $I=\{i_1,i_2,\cdots,i_n\}$ 是项的集合, 定义数据 D 是数据库事务的集合, 每个事务 T 是项的集合, $T\subseteq I$, 每个事务都有一个标识符 TiD。假设 X 是一个项集, 事务 T 包含 X 当且仅当 $X\subseteq T$, 关联规则是类似于 $X\rightarrow Y$ 的蕴含式, X 称为规则的先导, Y 称为规则的后继。

①关联规则类似于 $X\rightarrow Y$ 的蕴含式, 其中, $X\subset I$, $Y\subset I$, $X\bigcap Y=\varnothing$。

②关联规则类似于 $X\rightarrow Y$ 的蕴含式, 其支持度 Sup 是事务 D 中包含关联规则 $X\rightarrow Y$ 的百分比。

③关联规则类似于 $X\rightarrow Y$ 的蕴含式, 置信度 $Conf$ 是事务集 D 中同时包含 X 的事务和 Y 的事务的百分比。

可设 $MinSup$ 为最小支持度, 如果 $Sup\geqslant MinSup$, 则称项集为频繁项集。置信度 $Conf$ 是事务集 D 中同时包含 X 的事务和 Y 的事务的百分比, 那么规则 $X\rightarrow Y$ 支持度为 $P(X\bigcup Y)$, 置信度为 $P(X\,|\,Y)$。

(2) 加权关联规则与向下闭包特性

设 $I=\{i_1,i_2,\cdots,i_n\}$ 是项的集合, 那么在数据库事务集 D 中有 n 个项目, 为每个项都赋予一个权值与之对应, 那么它们的权值分别可以表示为 $W=\{w_1,w_2,\cdots,w_n\}$, 有 $0\leqslant w_i\leqslant 1$, $i=\{1,2,\cdots,n\}$, 指定最小加权支持度阈值 W_MinSup 和最小加权置信度阈值 $W\_MinConf$ 用于对加权关联规则的剪枝。

关联规则的挖掘归根结底是挖掘频繁集, 在传统的挖掘关联规则过程中, 由于项集数量庞大, 会产生数量巨大的子集, 利用频繁集向下闭包的性质, 能对规则进行剪枝, 从而过滤掉一些无关规则, 在传统的 Apriori 算法中, 如果{XY}和{YZ}不频繁, 那么{XYZ}和{YZW}也不频繁, 可以得出一个结论: 非频繁集的超集也是非频繁集。在算法执行过程中, 通过这一结论能够对非频繁集进行剪枝, 从而提高算法效率。

但在加权关联规则挖掘中, 该结论不成立, 换句话说, 加权频繁项集的任意子集可能不是加权频繁集; 非加权频繁项集的任意子集可能是加权频繁集[6]。故本文策略是挖掘出关联规则, 然后再证明一些规则是加权频繁的即可。

3.2 频次有效长度方法

为解决传统挖掘算法挖掘出大量冗余低价值规则的问题, 本文提出一种结合频次有效长度挖掘加权关联规则的算法, 在研究数据库中项目频次存在价值信息的基础上, 建立频次有效长度方法和权值设置方法, 通过这两种方法的结合, 进行稀有加权关联规则的挖掘。

(1) 滑动窗口技术与时序关联规则

本文挖掘加权关联规则算法的一个基础要素是要在具有时态约束的时序序列数据中进行。由于现实世界在不断地发展, 时间特性是反映现实世界信息的基本组成部分, 因此大多事务信息具有时间特性。又由于时序数据通常数量巨大, 且时间跨度相当长, 那么数据便有了“新”与“旧”之分, 为利用好这个时序序列数据, 将挖掘规则的过程放到滑动窗口进行。滑动窗

口的机制可以分为两个阶段: 初始阶段和滑动阶段。在初始阶段, 新数据不断地移进滑动窗口, 达到窗口长度值后便会停止移进数据, 没有其他数据移出; 在滑动阶段, 经典的滑动窗口采用移动覆盖的方法, 即随着窗口的滑动, 最旧的窗口内的数据被移出, 其他窗口数据向前移, 覆盖前一个窗口, 新数据移入到最末端的窗口。可简单理解为: 随着窗口的移动, "旧"数据移出, "新"数据移进。目的在于挖掘当前滑动窗口时间段内最具有价值的时序关联规则。

例如, 在传统的关联规则挖掘中对超市的数据集进行分析, 可以得到如"火鸡→南瓜馅饼(支持度＝0.0001, 置信度＝0.05)"的关联规则, 这意味着 0.01%交易事务中包含火鸡和南瓜馅饼, 而所有含有火鸡的交易中也有 5%含有南瓜馅饼。上述例子反映了三个问题: 上述规则不能视为一个突出的关联规则, 因为其支持度和置信度太低; 由于上述规则不突出, 只有把支持度和置信度调得足够低才能发现; 如果在夏天, 发起一项基于所发现的规则的促销活动, 比如一起购买火鸡和南瓜馅饼将享受10%的折扣, 毫无疑问商家会失败, 因为"火鸡"和"南瓜馅饼"在感恩节(冬季)假期前几周才会被大量销售。

从这个例子可以看出, 项目集在特定的时间间隔中频繁。因此, 有效地发现这些模式频繁出现的时间间隔至关重要。

(2) 频次有效长度方法

仔细思考上述"火鸡与南瓜馅饼"的例子, "火鸡"和"南瓜馅饼"在特定时间间隔中频繁, 也就是说"火鸡"和"南瓜馅饼"频繁的时间长度(周期)是一年。所以有依据猜测时序序列中数据存在周期性的联系。虽然可以猜测出数据库中的时序序列数据存在周期性, 但是周期性的对象不得而知, 故对数据库中的时序序列数据进行观察和实验, 发现事务数据中存在频次高低之分的类型项, 即频次出现越高的类型项重要性往往比其他类型项高, 从这一切入点入手, 统计每个事务中出现频次最高的类型, 再通过计算得出该类型的频次有效周期。

如果推荐预测范围仅仅是时序序列数据, 下一节点必定存在缺陷, 因为关联规则的后继可能有多个, 只选择其中一个来预测明显不合理。因此还要探索时序序列数据后续哪些节点内取得了最高的准确度。

频次有效长度是频次有效周期和预测节点长度的长度和, 也就是说频次有效长度将是滑动窗口的长度, 最后在频次有效长度内进行关联规则的挖掘, 能够发掘高价值的规则, 因此将探索到的新方法称为频次有效长度方法。

3.3　结合频次有效长度进行加权关联规则挖掘

(1) 算法描述

通常经典挖掘加权关联规则分为两大步骤: 挖掘出初始关联规则; 进行加权计算选取满足要求的规则。改进的加权关联规则算法基本步骤与经典算法一致。

算法 1: 频次有效长度方法挖掘非加权关联规则
输入: 数据库数据 *D*
输出: 非加权关联规则
Begin
①for *i*=1 to *D* do　//遍历数据库数据 *D*
②　for *j*=1 to sliding window(*sw*) do
　//遍历滑动窗口中数据
③　　Generate rules by new menthod
④　　Filter by their own sequence
　//用挖掘规则的后续序列过滤无关规则
⑤　End
⑥End

算法 1 首先在频次有效长度(滑动窗口)内挖掘出时序序列关联规则, 再通过自身的序列数据过滤掉部分无用规则。

算法 2: 在滑动窗口中挖掘加权关联规则
输入: 非加权关联规则; 最小加权支持度; 最小加权置信度; 数据库数据 *D*; 评分数据 *Rate*
输出: 满足最小加权支持度和最小加权置信度的加权关联规则
Begin
①for each *R*'∈*R* do　//遍历规则库中每个规则
②　for *s*=1 to *D*, *Rate* do //遍历时序序列数据与评分数据
③　　for *x*=1 to *U* do //遍历每个时序序列
④　　Calculate Frequency and Rating by using eq1 and eq2
　//用公式(1)和公式(2)计算平均频繁度与平均评分
⑤　End
⑥　End
⑦　Calculate the weight of transaction and the sum weight of transaction by using eq3
　//使用公式(3)计算事务权重以及事务权值总和
⑧　If *R*' in *sw* then
　//如果规则在滑动窗口中则进行下一步
⑨　Calculate the *wsup* and *wconf*

//计算加权支持度和加权置信度

⑩ End

⑪ If *R'.wSup*>=*W_MinSup* and *R'.wConf*>=*W_ MinConf*

//对比最小加权支持度和最小加权置信度

⑫ Add *R'* to queue //将满足要求的规则添加到列表中

⑬ End

⑭ End

算法 2 计算关联规则的加权支持度和加权置信度，选取高于最小加权支持度和最小加权置信度的关联规则。

(2) 挖掘频次有效长度内非加权关联规则

验证了数据库的时序序列数据中频次有效长度后，因为频次有效长度包含频次有效周期和预测节点长度，在频次有效周期长度内严格按照先后次序组配为关联规则的先导，将预测节点长度内的数据严格按照先后次序组配为关联规则的后继。这样可以挖掘出更多具有时态约束的关联规则。例如，有时序序列(*A*,*B*,*C*,*X*,*Y*)，假设恰好频次有效周期为 3，且序列为(*A*,*B*,*C*)，又如预测节点长度为 2，那么序列为(*X*,*Y*)。本文将序列(*A*,*B*,*C*)组配为关联规则的先导，即可以组配为：(*A*,*B*,*C*,*AB*,*AC*,*BC*,*ABC*)，将序列(*X*,*Y*)组配为(*X*,*Y*,*XY*)，由于严格按照先后次序组配原则，(*BA*，*CB*，*CBA*)不符合要求。于是可以得到关联规则(*A*→*X*，*A*→*Y*，*A*→*XY*，*B*→*X*，*B*→*Y*，*B*→*XY*，*C*→*X*，*C*→*Y*，*C*→*XY*，*AB*→*X*，*AB*→*Y*，*AB*→*XY*，*AC*→*X*，*AC*→*Y*，*AC*→*XY*，*BC*→*X*，*BC*→*Y*，*BC*→*XY*，*ABC*→*X*，*ABC*→*Y*，*ABC*→*XY*)。

(3) 权值的设计

笔者提出以下三个公式。

每种类型的平均频繁度(Frequency)指“每种类型数量和”除以“所有类型的总数量和”，如公式(1)所示。

$$Frequency = \frac{Number \cdot of \cdot every\_type}{Number \cdot of \cdot all\_type} \quad (1)$$

每种类型的平均评分(Rating)指“每种类型的评分和”除以“该类型总次数”，如公式(2)所示。

$$Rating = \frac{Number \cdot of \cdot every\_type\_rates}{Number \cdot of \cdot every\_type} \quad (2)$$

最终权值(Weight)指“每种类型的平均频繁度与每种类型的平均评分的乘积”作 * 运算，如公式(3)所示。

$$Weight = \{[Normalize(Frequency)] \times [Normalize(Rating)]\}^* \quad (3)$$

其中，*表示将最终权值放缩到[0,1]，为了使权值有意义且防止权值溢出[0,1]，或无限逼近[0,1]中的某个值，需通过放缩的运算，该方法是常用方法。

(4) 在滑动窗口中挖掘加权关联规则

由于在频次有效长度内已挖掘出非加权关联规则，现验证挖掘出的规则是否加权频繁，再将加权频繁的规则挑选出来。数据集如表 1 所示。

表 1 电影类型及评分

用户	类型	评分
1	*A*,*C*,*T*,*W*	3, 1, 5, 0.5
2	*C*,*D*,*T*	1 , 0.5 ,5
3	*A*,*C*,*W*	3 , 1 , 1.5
4	*A*,*C*,*D*,*W*	4, 1, 2, 1
5	*C*,*D*,*W*	1, 2, 1

根据公式(1)，*A* 类型的平均频繁度 =3/17 ≈ 0.18；同样地，*C*、*D*、*T*、*W* 类型的平均频繁度分别约为 0.29、0.18、0.12、0.24。根据公式(2)，*A* 类型的平均评分 =(3+3+4)/3 ≈ 3.33；同样地，*C*、*D*、*T*、*W* 类型的平均评分为 1、1.5、5、1。根据公式(3)，*A* 类型的最终权值为 0.18×3.33 ≈ 0.6；同样地，*C*、*D*、*T*、*W* 类型的最终权值经过规范化处理后分别约为 0.3、0.3、0.6、0.2。得到项目类型所对应的权值如表 2 所示。

表 2 项目类型所对应的权值

类型	权值
A	0.6
C	0.3
D	0.3
T	0.6
W	0.2

根据表 2，求每个类型项目权重和的平均值，可得用户序列平均权值如表 3 所示。

表 3 用户序列平均权值

用户	类型	计算过程	用户序列平均权值
1	*A*,*C*,*T*,*W*	(0.6+0.3+0.6+0.2)/4	0.43
2	*C*,*D*,*T*	(0.3+0.3+0.6)/3	0.4
3	*A*,*C*,*W*	(0.6+0.3+0.2)/3	0.37
4	*A*,*C*,*D*,*W*	(0.6+0.3+0.3+0.2)/4	0.35
5	*C*,*D*,*W*	(0.3+0.3+0.2)/3	0.27
	sum		**1.82**

对频次有效长度内挖掘出的关联规则进行加权。计算关联规则的加权支持度，将低于最小加权支持度阈值的关联规则进行剪枝。

如已挖掘出规则 $C \to T$ 、$AC \to W$ 和 $C \to W$，找出 <*CT*> 、<*ACW*> 、<*CW*> 所在的用户序列号，将其表示为：#1{<*CT*>,<1,2>} 、#2{<*ACW*>, <1,3,4>} 、#3{<*CW*>,<1,3,4,5>} 。其中 #1{<*CT*>, <1,2>} 中的 <1,2> 表示 <*CT*> 所在的用户序列号。

计算表 3 关联规则的加权支持度(*ws*):

<*CT*>的 *ws*=(0.43+0.40)/1.82=0.46

<*ACW*>的 *ws*=(0.43+0.37+0.35)/1.82=0.63

<*CW*>的 *ws*=(0.43+0.37+0.35+0.27)/1.82=0.78

因此可得到的结果如表 4 所示。

表 4　频繁集与其加权支持度

频繁集	CT	ACW	CW
加权支持度(*ws*)	0.46	0.63	0.78

若设置最小加权支持度为 0.6，那么 <*CT*> 所对应的规则 $C \to T$ 将被去除。

实际中用户序列长度很长，假设 <*CT*> 在用户序列中的位置可以表示在这样的一个时序序列中 *D*,*C*,*A*,*A*,*W*,*A*,*D*,*D*,*W*,*A*,*D*,*W*,*T*,*W*；由于 *C* 与 *T* 相对位置较远，甚至如果<*CT*>的加权支持度大于最小加权支持度的阈值，也不能盲目确定 $C \to T$ 规则适用。因此设计一个滑动窗口，使求用户序列平均权值的计算证明过程都在滑动窗口中进行。用户序列如表 5 所示，其中[ ]表示滑动窗口。

表 5　用户序列与滑动窗口

用户	类型
1	[*A*,*C*,*T*,*W*,*A*,*W*,*C*,*T*],*W*
2	[*D*,*T*, *T*,*W*,*A*,*W*,*C*,*A*],*W*,*C*
3	[*W*, *T*, *T*,*W*,*A*,*W*,*A*,*T*],*W*,*A*,*C*
4	[*A*,*C*,*D*,*W*,*T*,*A*,*C*,*W*]
5	[*C*,*D*,*W*]

可求滑动窗口中序列平均权值，得到滑动窗口中的用户序列平均权值，如表 6 所示。假设有规则 $TWA \to C$，该规则的频繁集为 <*TWAC*>，<*TWAC*> 所在的用户序列为 <1,2>，将其表示为 #1{<1,2>, <*TWAC*>}，该关联规则的加权支持度 (*ws*) =(0.45+0.51)/2.1=0.46；如果设置最小加权支持度为 0.4，那么该规则会被保留，如果设置最小加权支持度为 0.5，那么该规则会被删除。需要注意的是频繁集 <*WTAC*> 不能和 <*TWAC*> 等价，因为挖掘的是时序序列模式中的加权关联规则。

表 6　滑动窗口中的用户序列平均权值

用户	类型	滑窗中的用户序列平均权值
1	[*A*,*C*,*T*,*W*,*A*,*W*,*C*,*T*],*W*	0.45
2	*C*,[*D*,*T*, *T*,*W*,*A*,*W*,*C*,*A*],*W*	0.51
3	*A*,*C*,[*W*, *T*, *T*,*W*,*A*,*W*,*A*,*T*],*W*	0.56
4	[*A*,*C*,*D*,*W*,*T*,*A*,*C*,*W*]	0.38
5	[*C*,*D*,*W*]	0.20
	sum	**2.10**

4　实验对比分析

4.1　Apriori 算法挖掘具有时态约束的关联规则分析

利用 Apriori 算法在时序序列中挖掘时态约束的非加权关联规则，并以非加权关联规则进行推荐实验。推荐策略是一旦规则的先导匹配上时序序列中的某一段序列，用该序列的后一个邻近节点作为预测对象与规则的后继进行匹配，相同即为命中。

覆盖度指推荐的项目集合占总项目的比例，为了更细致地描述推荐系统发掘长尾的能力。推荐测试的准确度与覆盖度分别如表 7 和表 8 所示。

表 7　Apriori 各区间准确度

支持度 / 置信度	(0.20,0.25]	(0.25,0.30]	(0.30,0.35]	(0.35,0.40]
(0.5,0.6]	0.0378	0.0319	0.0579	0.0287
(0.6,0.7]	0.0469	0.0432	0.0599	0.0433
(0.7,0.8]	0.0748	0.0802	0.0967	0.0830
(0.8,0.9]	0.1020	0.1174	0.0935	0.0667
(0.9,1.0]	0.0462	0.0272	0.0090	0.0066

表 8　Apriori 各区间覆盖度

支持度 / 置信度	(0.20,0.25]	(0.25,0.30]	(0.30,0.35]	(0.35,0.40]
(0.5,0.6]	0.0031	0.0019	0.0012	0.0009
(0.6,0.7]	0.0091	0.0035	0.0026	0.0018
(0.7,0.8]	0.0190	0.0071	0.0039	0.0024
(0.8,0.9]	0.0543	0.0196	0.0059	0.0025
(0.9,1.0]	0.0826	0.0185	0.0029	0.0011

4.2　扩大预测范围

由于关联规则的后继有可能有多个，只选择其中

一个推荐不合理。如有规则($A \rightarrow B,C$), 先导匹配上某序列后预测范围若只为(B), 此时的预测方法不知如何进行。采用经典的 Apriori 算法在时序序列中挖掘时态约束的关联规则后, 再扩大预测范围, 通过在训练集中的实验, 发现扩大预测范围后, 具有最高预测命中率的范围为预测节点后 5 个节点内。后续各个节点位置的命中率如表 9 所示。因为后面 5 个节点内的准确度最高, 所以选取后面 5 个节点内作为预测范围。

表 9　后续各个节点位置的命中率

后续节点位置	命中率	后续节点位置	命中率
1	0.3750	6	0.3701
2	0.3722	7	0.3651
3	0.3725	8	0.3685
4	0.3704	9	0.3695
5	0.3853	10	0.3680

扩大预测范围后推荐准确度与覆盖度如表 10 和表 11 所示。

表 10　扩大预测范围后各区间准确度

支持度 / 置信度	(0.20,0.25]	(0.25,0.30]	(0.30,0.35]	(0.35,0.40]
(0.5,0.6]	0.0432	0.0455	0.0554	0.0308
(0.6,0.7]	0.1121	0.1079	0.0560	0.0975
(0.7,0.8]	0.2124	0.1594	0.1142	0.0994
(0.8,0.9]	0.1832	0.1730	0.1270	0.0903
(0.9,1.0]	0.0969	0.0534	0.0172	0.0156

表 11　扩大预测范围后各区间覆盖度

支持度 / 置信度	(0.20,0.25]	(0.25,0.30]	(0.30,0.35]	(0.35,0.40]
(0.5,0.6]	0.0047	0.0022	0.0018	0.0008
(0.6,0.7]	0.0162	0.0127	0.0049	0.0029
(0.7,0.8]	0.0310	0.0122	0.0065	0.0042
(0.8,0.9]	0.0761	0.0236	0.0097	0.0030
(0.9,1.0]	0.1285	0.0301	0.0054	0.0013

4.3　本文方法挖掘加权关联规则的分析

在对关联规则进行挖掘前, 需验证在训练集中频次有效周期的值, 为此截取不同长度的时序序列, 计算并统计每个长度所对应的误差率如表 12 所示, 发现有效长度为 8 个节点长度时误差率最低。预测节点长度已证明是 5 个节点长度, 由此可以得出本实验的频次有效长度(13)是频次有效周期(8)与预测节点长度(5)的和, 按照新的挖掘方法在频次有效长度(滑动窗口)内进行加权关联规则的挖掘。将规则按照不同加权支持度形成区间, 计算各个区间的推荐准确度和覆盖度, 如表 13 和表 14 所示。

表 12　频次有效周期与误差率

频次有效周期	误差率	频次有效周期	误差率
3	0.3201	8	0.2449
4	0.2995	9	0.2658
5	0.2831	10	0.2576
6	0.2769	11	0.2694
7	0.2769	12	0.2603

表 13　新方法各加权支持度区间的准确度

区间	准确度	区间	准确度	区间	准确度
(0.10,0.15]	0.69	(0.40,0.45]	0.38	(0.70,0.75]	0.45
(0.15,0.20]	0.55	(0.45,0.50]	0.40	(0.75,0.80]	0.07
(0.20,0.25]	0.53	(0.50,0.55]	0.42	(0.80,0.85]	0.15
(0.25,0.30]	0.49	(0.55,0.60]	0.20	(0.85,0.90]	0.11
(0.30,0.35]	0.52	(0.60,0.65]	0.14	(0.90,0.95]	0.10
(0.35,0.40]	0.43	(0.65,0.70]	0.23	(0.95,1.00]	0.01

表 14　新方法加权支持度各区间的覆盖度

区间	覆盖度	区间	覆盖度	区间	覆盖度
(0.10,0.15]	0.40	(0.40,0.45]	0.35	(0.70,0.75]	0.31
(0.15,0.20]	0.41	(0.45,0.50]	0.33	(0.75,0.80]	0.31
(0.20,0.25]	0.36	(0.50,0.55]	0.22	(0.80,0.85]	0.27
(0.25,0.30]	0.38	(0.55,0.60]	0.34	(0.85,0.90]	0.22
(0.30,0.35]	0.37	(0.60,0.65]	0.30	(0.90,0.95]	0.25
(0.35,0.40]	0.36	(0.65,0.70]	0.31	(0.95,1.00]	0.08

准确度最高加权置信度区间的所有规则的加权置信度都高于 0.30。当加权置信度为 0.40 时准确度最高。设置不同最小置信度阈值所对应的准确度如表 15 所示。

表 15　最小加权置信度阈值所对应的准确度

最小加权置信度阈值	准确度	最小加权置信度阈值	准确度
0.40	0.69	0.70	0.57
0.50	0.67	0.80	0.53
0.60	0.65	0.90	0.48

4.4　对比实验

各个方法在准确度方面的对比如图 1 所示, 包括单区间最高准确度的比较, 所有区间平均准确度的比较; 各个方法在覆盖度方面的对比如图 2 所示, 包括单区间最高覆盖度的比较, 还有所有区间平均覆盖度的比较。可以发现, 本文所提挖掘方法在推荐准确度

和覆盖度方面都较以往有较大提升。

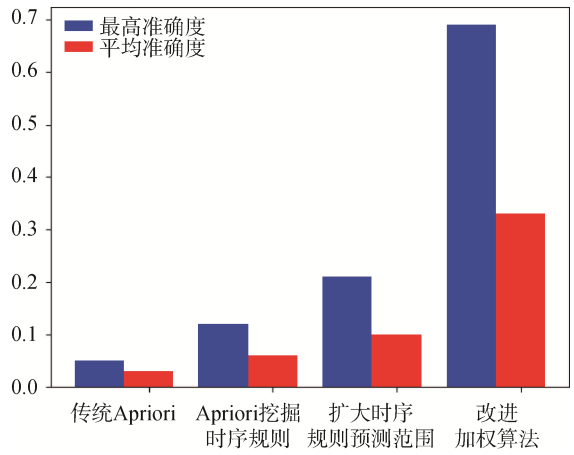

图 1　准确度方面对比

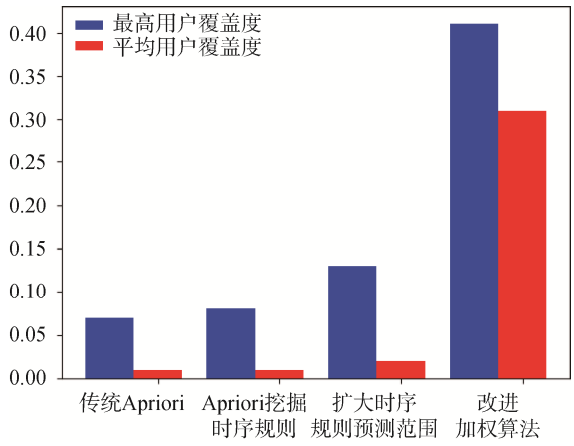

图 2　覆盖度方面对比

通过观察表 10 可以发现，支持度和置信度分别在(0.20,0.25]和(0.7,0.8]区间时，最高准确度高达 21.24%，相比于表 7 的最高准确度 11.74%，提升 81%。通过观察表 11 可以发现，支持度和置信度分别在(0.20,0.25]和(0.9,1.0]区间时，最高覆盖度达 12.85%，相比于表 8 的最高覆盖度 8.26%，提升 56%。表 10 相比于表 8，可以发现大多数区间所对应的准确度都有飞跃性的提升。准确度方面，表 10 整体准确度比表 7 提升 64%；支持度与置信度分别在(0.20,0.25]和(0.7,0.8]区间时，表 10 的准确度相比于表 7 提升了 1.8 倍；支持度与置信度分别在(0.30,0.35]和(0.5,0.6]区间时，表 10 的准确度相比于表 7 却下降 4%。覆盖度方面，表 11 整体覆盖度比表 8 提升 55%；支持度与置信度分别在(0.25,0.30]和(0.6,0.7]区间时，表 11 的覆盖度相比于表 8 提升了 2.6 倍；支持度与置信度分别在(0.35,0.40]和(0.5,0.6]区间时，表 11 的覆盖度相比于表 8 却下降 11%。

通过表 13 和表 14 可见，加权支持度在(0.10,0.15]区间命中最高，且覆盖度排第二，规则的数量随着加权支持度的增加而减少。加权支持度在区间(0.10,0.15]时，最高准确度达 69.0%，相比于表 10 的最高准确度 21.24%，提升 225%。加权支持度在区间(0.15,0.20]时，最高覆盖度达 41.0%，相比于表 11 的最高覆盖度 12.85%，提升 219%。另外表 13 在整体准确度方面比表 10 提升 21 倍；表 14 在整体覆盖度方面比表 11 提升 13.7 倍。

本文方法的准确度与其他已有方法对比如图 3 所示。可以发现，在本文的新加权关联规则挖掘方法下，推荐的准确度高达 69.0%，与基于 FP-树的加权关联规则算法[8]、改进的 MINWAL 算法[9]、基于演化规则集的推荐算法[10]和基于行为和评分相似性的关联规则群推荐算法[11]在准确度方面相比有优势，说明本文所提算法具有有效性与可行性。

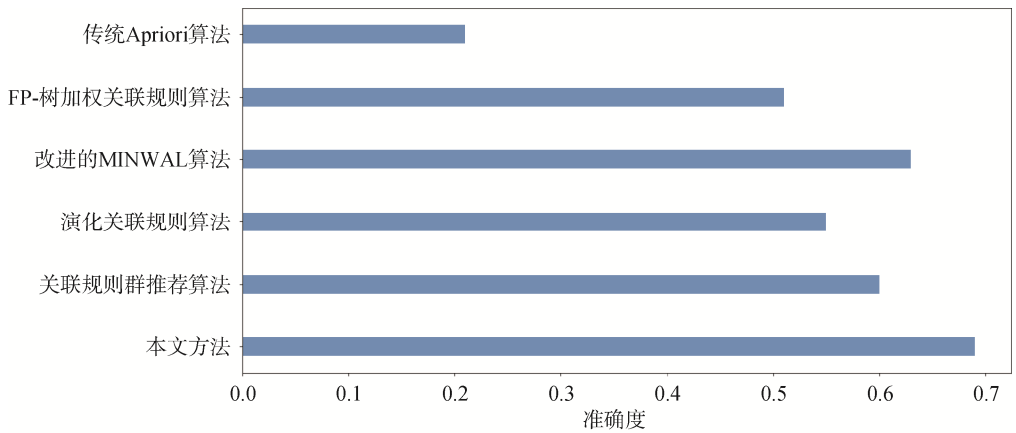

图 3　本文方法的准确度与已有方法对比

5 结 语

本文提出一种改进的加权时序关联规则挖掘算法, 为体现各个项目独立的重要性, 对项目引入权值, 但由于引入权值, 导致频繁项目集的子集不再频繁, 故采取先挖掘出规则, 再证明并选出具有加权频繁集特性的规则。由于时间特性是反映现实世界信息的基本组成部分, 因此大多事务信息具有时间特性。改进的加权时序关联规则挖掘算法所挖掘出的加权时序关联规则在准确度和覆盖度方面表现优异。本文旨在挖掘出隐藏的高价值规则, 对算法执行时间复杂度有所忽视, 未来将在保证准确度的同时, 兼顾算法执行的时间复杂度, 对算法进行优化。

参考文献:

[1] Khan M S, Muyeba M, Coenen F. A Weighted Utility Framework for Mining Association Rules[C]// Proceedings of the 2nd UKSIM European Symposium on Computer Modeling and Simulation. 2008: 87-92.

[2] Forsati R, Meybodi M R. Effective Page Recommendation Algorithms Based on Distributed Learning Automata and Weighted Association Rules[J]. Expert Systems with Applications, 2010, 37(2): 1316-1330.

[3] Zhai Y, Wang L, Wang N. Efficient Weighted Association Rule Mining Using Lattice[C]// Proceedings of the 26th Chinese Control and Decision Conference. 2014: 4913-4917.

[4] Ouyang W. Mining Weighted Rare Association Rules Using Sliding Window over Data Streams[C]// Proceedings of the 2016 International Conference on Computer Science and Electronic Technology. 2016: 116-119.

[5] 李成军, 杨天奇. 一种改进的加权关联规则挖掘方法[J].计算机工程, 2010, 36(7): 55-57. (Li Chengjun, Yang Tianqi. Improved Weighted Association Rules Mining Method[J]. Computer Engineering, 2010, 36(7): 55-57.)

[6] 欧阳为民, 郑诚, 蔡庆生. 数据库中加权关联规则的发现[J]. 软件学报, 2001, 12(4): 612-619. (Ouyang Weimin, Zheng Cheng, Cai Qingsheng. Discovery of Weighted Association Rules in Databases[J]. Journal of Software, 2001, 12(4): 612-619.)

[7] Malarvizhi S P, Sathiyabhama B. Frequent Pagesets from Web Log by Enhanced Weighted Association Rule Mining[J]. Cluster Computing, 2016, 19(1): 1-9.

[8] 王涛伟, 任一波. 基于加权关联规则的个性化推荐研究[J]. 计算机应用与软件, 2008, 25(8): 242-244. (Wang Taowei, Ren Yibo. Study on Personalized Recommendation Based on Weighted Association Rule[J]. Computer Applications and Software, 2008, 25(8): 242-244.)

[9] 王斌, 丁祥斌. 一种基于 BUC 的水平加权关联规则挖掘算法[J]. 计算机应用与软件, 2008, 25(12): 112-115. (Wang Bin, Ding Xiangbin. A BUC-Based Mining Algorithm for Horizontal Weighted Association Rules[J]. Computer Applications and Software, 2008, 25(12): 112-115.)

[10] 龙舜, 蔡跳, 林佳雄. 一个基于演化关联规则挖掘的个性化推荐模型[J]. 暨南大学学报: 自然科学与医学版, 2012, 33(3): 264-267. (Long Shun, Cai Tiao, Lin Jiaxiong. A Personalized Recommendation Model Based on Evolving Association Rule Mining[J]. Journal of Jinan University: Natural Science & Medicine Edition, 2012, 33(3): 264-267.)

[11] 张佳乐, 梁吉业, 庞继芳, 等. 基于行为和评分相似性的关联规则群推荐算法[J]. 计算机科学, 2014, 41(3): 36-40. (Zhang Jiale, Liang Jiye, Pang Jifang, et al. Behavior and Score Similarity Based Algorithm for Association Rule Group Recommendation[J]. Computer Science, 2014, 41(3): 36-40.)

作者贡献声明:

张勇: 进行实验, 采集、清洗和分析数据, 论文起草;
李树青: 设计研究方案, 论文最终版本修订;
程永上: 提出研究思路。

利益冲突声明:

所有作者声明不存在利益冲突关系。

支撑数据:

支撑数据由作者自存储, E-mail: zhangyong207318@163.com。
[1] 张勇. 时序序列数据. csv. 实验数据.
[2] 张勇. 时序序列对应的评分数据. csv. 实验评分数据.

收稿日期: 2018-09-08
收修改稿日期: 2018-11-02

Mining Algorithm for Weighted Association Rules Based on Frequency Effective Length

Zhang Yong　Li Shuqing　Cheng Yongshang
(School of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210046, China)

Abstract: [Objective] This paper analyzes the differences in the importance of database items, aiming to address the issues of traditional association mining algorithm with redundant and worthless rules. **[Methods]** On the sequence with temporal constraints, we explored the non-weighted association rules with the frequency effective length and the weighting methods. Then, we used sliding window technique to study the rare weighted association rules on the time series. **[Results]** The accuracy of the prediction made by the proposed method increased to 69% from 62%. **[Limitations]** The mining algorithm took long time to extract the needed rules due to the sliding windows and the large number of rules generated. **[Conclusions]** The association rules of weighted time series improve the accuracy of recommendation, which also provides new directions for research method on association rules.

Keywords: Data Mining　Association Rules　Frequency Length　Sliding Window

第二届“数据分析与知识发现”学术研讨会成功召开

2019 年 7 月 10-11 日, 第二届“数据分析与知识发现”学术研讨会(DAKD 2019)在兰州成功召开。本次会议由中国科学院文献情报中心主办, 中国科学院兰州文献情报中心和《数据分析与知识发现》编辑部联合承办。中国科学院文献情报中心刘会洲主任出席会议并作开幕致辞。来自 75 家高校及科研院所, 跨计算机科学、情报科学、管理科学、经济学、行为科学、心理学等众多领域的近 300 名专家和代表参加了本次会议。

知识环境的改变对知识服务提出新的需求和新的挑战。DAKD 2019 专家报告覆盖了从 Data 到 Information, 再到 Intelligence, 直至 Solution 的丰富内容。中国科学院计算机研究所王元卓研究员作“大数据与开放知识计算”报告、清华大学邢春晓教授作“大数据驱动的医疗健康知识管理和决策”报告、复旦大学肖仰华作“当知识图谱‘遇见’深度学习”报告、中国科学院兰州文献情报中心曲建升研究员作“当前战略情报工作的挑战与机会”报告, 内容既聚焦前沿, 又深入需求, 揭示了在构建面向未来的知识服务系统中资源、方法和用户缺一不可。北京大学张岩教授、中国科学院计算机网络信息中心沈志宏研究员、南京大学程龚教授分享了大数据领域新的技术架构和技术思路。中国科学院文献情报中心张智雄研究员、乐小虬研究员、南京理工大学章成志教授针对文本抽取领域分享了最新的研究成果。武汉大学张李义教授、北京理工大学牛振东教授、中国科学院兰州文献情报中心祝忠明研究员、西南大学贾韬教授、南京理工大学李千目教授、华中师范大学王伟军教授、大连理工大学金博教授、南京农业大学王东波教授、南京大学丁晓蔚教授等展示了大数据时代精彩的行业应用案例, 针对不同领域的应用需求, 提出创新性的原理、指标、算法和实施机制, 描绘了跨界合作的前景。中国科学院文献情报中心张晓林研究员做会议闭幕总结, 剖析了重构知识服务生态系统的难点痛点问题; 阐述了用户需求驱动、嵌入用户流程、融合用户参与的构建思路; 提出打造多元多方多领域甚至跨界的深度合作将成为常态。

本次会议征文得到了国内高校及科研院所的大力支持。共计收到征文近百篇, 评选出优秀论文 25 篇, 将推荐至《数据分析与知识发现》期刊发表。

(本刊讯)