

兴趣度在增量的关联规则挖掘中的研究

向哲, 林国龙, 杨斌

(上海海事大学, 上海 200135)

摘要:一般的关联规则发现算法使用的都是支持度、置信度框架。但是在增量的数据挖掘过程中, 该类算法却需要不断改变支持度、置信度, 使得算法本身效率下降, 并缺乏可说服力, 比如 Apriori 算法。为了解决该类问题, 使用兴趣度框架对增量的数据进行了关联规则挖掘, 比较了基于支持度、置信度框架的算法(如 Apriori, FUP 算法)和基于兴趣度的算法之间的优缺点。试验结果表明: 兴趣度能够有效地筛选关联规则, 在进行增量的数据挖掘的情况下得到的关联规则总是小于等于支持度、置信度(Apriori)算法挖掘出的规则。

关键词:关联规则; FUP 算法; Apriori 算法; 兴趣度

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2009)10-0033-04

Interestingness Research of Association Rules in Incremental Mining Data

XIANG Zhe, LIN Guo-long, YANG Bin

(Shanghai Maritime University, Shanghai 200135, China)

Abstract: The general association rules discovery algorithm is using in the framework of confidence and support. However, in incremental data mining process, such algorithms are required support and confidence to constantly changing. Makes inefficient algorithm itself and the lack of persuasive, such as Apriori algorithm. In this paper, in order to solve such problem, use interestingness framework in incremental degrees of data mining association rules, based on the comparative degree of support, confidence in the framework of the algorithm (such as Apriori, FUP algorithm) and the algorithm based on the degree of interest between the gifted disadvantage. The experimental results show that: Interestingness be able to effectively filter association rules, the association rules in incremental data mining to be always less than or equal degree of support, confidence level (Apriori) algorithm excavated rules.

Key words: association rules; FUP algorithm; Apriori algorithm; interestingness

0 引言

关联规则挖掘是研究数据潜在有趣模式的一项重要方法, 它揭示数据库各属性之间的内在关联, 向用户提供大量的有趣规则。实际上, 数据仓库随着时间的推移, 会增加、删减、更新数据, 也就是一般所说的增量数据库。因此, 相关的关联规则算法也就应运而生了。至今已有了很多高效的关联规则挖掘算法, 如 Apriori, AIS, DHP, Partition^[1]和抽样算法, 但这些算法大多针对静态的数据, 实际应用中目标数据是随时间不断变化的, 因此, 迫切需要设计高效的算法来更新、维护和

管理已被挖掘的关联规则。

在增量关联规则算法的研究中, 人们先后研究了基于 Apriori 的 FUP、FUP2 算法; UWEP 算法; 基于 FP-Tree 的 FIUA 算法; 基于 DIC 的 IDIC-M、IDIC-I 算法, IUA 及 PIUA 算法^[1,2], 这些算法都是基于支持度、置信度框架的。

兴趣度的研究领域里, 应用最广的是 PS 模型的兴趣度, 为了解决兴趣度的值域问题, 还先后引入了准确度(Accuracy)^[3], S—指数模型^[4,5]。另外, 为了更全面地考虑兴趣度的权值还加入了带熵值的兴趣度公式。

为了更好地理解文中所要阐述的内容, 来考虑下面一个问题:

以往的算法, 比如 FUP^[1]算法(该算法也是基于 Apriori 算法的), 考虑的是支持度和置信度阈值, 在增量的数据仓库中, 这一类算法都需要不断地更改阈值, FUP 算法虽然能够计算出适当的支持度、置信度, 但

收稿日期: 2009-02-26; 修回日期: 2009-05-05

基金项目: 上海市引进技术吸收与创新年度计划资助项目(07XI-058); 上海市能力建设资助项目(071705107, 0817 0511300)

作者简介: 向哲(1984-), 男, 广西桂林人, 硕士研究生, 研究方向为物流信息化; 林国龙, 教授, 研究方向为物流信息化; 杨斌, 副教授, 研究方向为物流信息化。

它依然要对该阈值进行修改。而通过兴趣度的挖掘,我们发现只有真正相关的项目才有可能被挖掘出来,它们的相关度不会像 FUP 算法一样受数据的影响,这一点下面也会进行证明。

1 关联规则的问题描述

1.1 介绍兴趣度

一般的支持度^[1] $\text{Support}(A \cup B)$, $\text{Confidence}(A \cup B)$ 。

兴趣度公式为^[1]:

$$\text{Interest}(A \cup B) = P(AB)/P(A)P(B) \quad (1)$$

该算法的引入主要考虑了概率论中的相关性,也就是说只有两者真正相关才能被挖掘出来,这恐怕才是数据挖掘的初衷。

1.2 引言中描述的问题

接着,来说明引言中提到的问题。考虑下面的问题(FUP 算法的优缺点^[6]):

给定最小支持度和最小置信度,当一个新的数据集 db 添加到旧的数据库 DB 中时,如何生成 $DB \cup db$ 中的关联规则。

用数学语言来描述增量关联规则的问题^[2,6]: 设 L 是原始数据库 DB 的所有频繁项集的集合, s 是最小支持度, D 为数据库 DB 中的事务数量。对 L 中每一个频繁项集 $X \in L$, 它的支持数为 X_{support} , 即在 DB 中包含了 X 的事务数量。设更新后的增量数据集 db 的事务总数为 d , 最小支持度 s 保持不变, 一个项集 X 在更新后的数据库 $DB \cup db$ 中频繁的充要条件是: X 在 $DB \cup db$ 中的支持度不小于 s , 即

$$X_{\text{support}} \geq s \times (D + d)/D \quad (2)$$

这是经典的增量关联规则算法——FUP 算法。FUP 基于 Apriori 并假设保存了已有频繁项集和支持度, 其运行效率要比重新运行 Apriori 高得多^[2]。但也出现了一些问题: 一是由于 FUP 算法需要分别从原数据集和增量数据集中计算出候选项集^[5], 需计算候选项集数目非常多, 特别是候选 2-项集; 二是它需要重复扫描整个更新后的数据库的次数跟最长频繁项集长度相等, 对候选项目集进行模式匹配, 并没有降低 I/O 代价^[7]; 三是算法对新增项目不敏感^[6]。

实际上造成这几点的原因主要还是因为虽然它能够计算出新的支持度、新的置信度, 但它还是要对原来的支持度进行修改, 降低了算法效率^[7]。引入兴趣度, 虽然算法本身还是要依托支持度的计数, 筛选, 但不管支持度如何改变项目之间的相关性由于决定于两者出现的概率的大小, 不会像支持度那样, 受之前设定的兴趣度的影响。

2 增量数据中兴趣度的相关证明

上一节中作出了一个推论就是兴趣度决定于两者出现概率的大小, 兴趣度不会像支持度那样, 在增量数据中受之前设定的支持度的影响。把这个问题作为一个推论来进行证明, 该推论如下:

推论: 增量数据库中, 兴趣度(PS 模型)的偏移(变化)度小于支持度的偏移度。

将推论用数学语言描述一遍: 设 DB 中 A 满足最小支持度 S , 记为 $P(A)$ 。 B 为大于最小支持度的项目, 其概率为 $P(B)$, 设在 db 中 A 若要继续满足最小支持度则需要其出现的概率大于 $s \times (D + d)/D$, 记为 $P(A')$, B 的概率为 $P(B')$, 设 DB, db 中的兴趣度分别为 $I(AB)$ 和 $I(A'B')$ 。

现在要证明的是 $I(AB)$ 和 $I(A'B')$ 之间的变化小于 $P(A)$ 和 $P(A')$ 之间的变化。

证明:

依据兴趣度(PS 模型^[8]) 公式可以得出:

$$\begin{aligned} I(AB) &= P(AB)/P(A)P(B) \\ I(A'B') &= P(A'B')/P(A')P(B') \end{aligned} \quad (3)$$

要证明推论就需要证明

$$\frac{I(AB)}{I(A'B')} \leq \frac{P(A)}{P(A')} \quad (4)$$

开始对不等式左边进行变型, 由公式(2) 得:

$$\text{因为 } P(A) = S, P(A') = S(D + d)/D$$

$$\text{所以 } \frac{P(A)}{P(A')} = \frac{S}{S(D + d)/D} = \frac{D}{D + d} \quad (5)$$

不等式右边为

$$\begin{aligned} \frac{P(AB)/P(A)P(B)}{P(A'B')/P(A')P(B')} &= \\ \frac{D}{D + d} \times \frac{P(B)}{P(B')} \times \frac{P(AB)}{P(A'B')} \end{aligned} \quad (6)$$

又 $P(B)$ 为大于原先的最小支持度的项目的概率, $P(B')$ 大于增量后支持度的概率。

又 $P(AB)$, $P(A'B')$ 分别是 $P(A)$, $P(A')$ 的真子集且又要被筛选出来(即又要大于等于 $P(A)$, $P(A')$)。

$$\text{因此 } \frac{P(AB)}{P(A'B')} = \frac{D}{D + d}$$

所以原式:

$$\frac{I(AB)}{I(A'B')} = \left(\frac{D}{D + d}\right)^2 \times \frac{P(B)}{P(B')} \quad (7)$$

同时 $P(B)$, $P(B')$ 若要被挖掘出来, $\frac{P(B)}{P(B')}$ 必然小于 1, 且 $\frac{D}{D + d} \leq 1$ 。

$$\text{因此 } \frac{I(AB)}{I(A'B')} = \left(\frac{D}{D + d}\right)^2 \times \frac{P(B)}{P(B')} \leq \left(\frac{D}{D + d}\right)^2 \quad (8)$$

由公式(7), (8) 明显可得

$$\frac{I(AB)}{I(A'B')} = (\frac{D}{D+d})^2 \times \frac{P(B)}{P(B')} \leq (\frac{D}{D+d})^2 \leq \frac{D}{D+d}$$
$$= \frac{P(A)}{P(A')} \tag{9}$$

故获证。

因此,增量数据库中,兴趣度(PS 模型)的偏移度小于支持度的偏移度,这就说明当数据增加时,基于兴趣度挖掘出的关联规则的变化会小于基于支持度置信度挖掘出的关联规则的变化,这种变化主要体现在关联规则的数目方面,下面将用仿真来进行说明。

3 三种算法的仿真

文中所用的仿真程序均用 T-SQL 编写,采用 SQL2005 数据库,文中的样例数据集是某商店一天的商品销售数据,分为五项:牛奶、咖啡、白糖、饮料、茶叶。为了简单起见,在进行数据挖掘的时候对数据进行了处理。使用英文字符来代替事务项目,字符 a 代表了牛奶,b 代表了咖啡,c 代表了白糖,d 代表了饮料,e 代表了茶叶,整理后见表 1。

表 1 原始数据集和增量数据集
(上为原始数据,下为增量数据)

Buy1	1. bd 2. bc 3. abc 4. ad 5. bcd 6. bc 7. cd 8. acd 9. bc 10. bcd 11. a 12. bce
Buy2	1. bd 2. bc 3. abc 4. ad 5. bcd 6. bc 7. cd 8. acd 9. bc 10. bcd 11. a 12. bce13. bd 14. bc 15. abc 16. ad 17. bcd 18. bc 19. cd 20. acd 21. bc 22. bcd 23. a 24. bce

先分别描述 Apriori 算法和 FUP 算法的步骤。

●Apriori 算法步骤如下^[8]:

第一步:对数据表 message 进行扫描,计算出所有出现的项目及其个数,生成 c_1 。

第二步:基于最小支持度对刚生成的表进行筛选,得到频繁项集 - 1。

第三步:对频繁项集 - 1(用 L_1 表示)求 Apriori - gen 求出候选项集 c_2 。

第四步:基于最小支持度生成 L_2 ,之后以次类推。

第五步:迭代结束后,通过最小置信度找出关联规则。

●FUP 算法的步骤如下^[5]:

第一步:保存了已有频繁项集和支持度,计算 $X_{support}$ 。

第二步:其余步骤参照 Apriori 算法。支持度变成 $X_{support}$ 。

兴趣度的步骤几乎与 Apriori 算法一样,只是在最后判定输出的时候不一样,所以这里不再赘述。各算法门限阈值见表 2。

表 2 各门限的具体数值

	Apriori 算法	FUP 算法	兴趣度算法
支持度计数	3,3	3,6 ($s \times (D + d)/D$)	
置信度	0.2,0.2	0.2	
兴趣度			0.1,0.1

Apriori 算法前后输出关联规则比较见表 3,表 4。

表 3 原样例数据集挖掘的规则

	关联规则
2 阶	$b = > cd, c = > bd, bc = > d$

表 4 增量样例数据集挖掘的规则

	关联规则
1 阶	$a = > b, a = > c, a = > d, b = > a, c = > a, d = > a$
2 阶	$a = > bc, b = > ac, b = > cd, c = > ab, c = > bd$ $bc = > a, bc = > d$

挖掘的结果如上,期间未修改支持度、置信度。FUP 算法试验结果见表 5,表 6。

表 5 FUP 挖掘样例数据集的规则

ID	关联规则
1	$b = > cd, c = > bd, bc = > d$

表 6 FUP 挖掘样例数据集计算的支持度

ID	计算出的支持度
1	6

兴趣度算法结果见表 7。

表 7 兴趣度挖掘的关联规则结果

ID	兴趣度
1	$b = > bc, b = > bcd$

增量前后挖掘的规则完全是一样的。

综上所述,分析了三种算法之间结果上的不同,值得注意的是,增量的数据 db 完全是 DB 的复制,从结果上看,Apriori 算法前后挖掘出的关联规则差距非常大,但从道理上来说,同样的数据包含的规则也应该是相同的,这也就说明使用 Apriori 算法来进行增量挖掘是一种错误的选择。

4 仿真结论

从结果上看,可以对支持度、置信度框架的算法如 Apriori 算法和 FUP 算法,以及对兴趣度框架的算法作出以下几点结论:

- 第一,支持度、置信度框架的优点是:算法过程简单,思想清晰,能够进行基本的项目筛选,数据挖掘。
- 第二,支持度、置信度框架缺点在于:阈值的设定对数据样例的属性依赖很高。如果加入数据,那么支持度、置信度就需要不断增加才能挖掘出适量的关联

规则。

第三,由实验可见,支持度、置信度框架中如果一个项目大量出现,那么它非常容易成为候选项目集,包括它的 2—项集,3—项集也会大量,并容易被当作关联规则而被挖掘出来,这是否完全正确呢?值得考虑。

总之,兴趣度的衡量框架优于支持度、置信度框架的原因在于计算了各个项目之间相互条件下出现的概率,对单一大量出现的项目不敏感(当然如果它引起其他项目大量出现就会敏感了)。所以,笔者认为,关联规则挖掘时更加应当考虑兴趣度。

5 结束语

文中利用比较直观的方法分析了支持度置信度框架下的两种算法(Apriori 算法和 FUP 算法)和兴趣度框架下兴趣度算法之间的不同点,并作出了结论。事实证明,兴趣度能够更好地体现数据挖掘的初衷。文中还在第 2 章证明了一个推论,能够说明为什么在增量数据库中,支持度和兴趣度都变化的情况下,兴趣度的变化要小于支持度的变化,从而使挖掘出的新挖掘出的关联规则小于支持度挖掘出的关联规则。

由于兴趣度框架还在发展阶段,有很多体系还不完善,比如,虽然兴趣度挖掘出的关联规则少于支持度置信度挖掘出的关联规则,但是否这些规则更能说明问题呢,虽然针对该问题举了一个例子,但要从理论上证明起来,还有待进一步研究。

(上接第 32 页)

从图中可看出,节点有效时间比率 t 较小时,随着 n 的增加,消息成功传输的概率也逐渐增加。因此,使用组代理机制能够使得在节点不可靠的情况下,通过增加组成消息传输代理的节点的数目,提高代理传递消息的可靠性。

6 结束语

文中提出的组代理机制利用多个节点共同转发消息,在提高了代理传递消息的可靠性的同时,也保证了消息传递的正确性。组代理机制克服了采用单个节点传递消息时由于单个转发节点失效而影响消息传递的缺点,在部分组代理节点失效的情况下,消息仍能得到传输。

参考文献:

- [1] Wu Jie. 分布式系统设计[M]. 高传善,等译. 北京:机械工业出版社,2001.
- [2] Sun Microsystems. Java Message Service(JMS)Specification

参考文献:

- [1] Tansel A U, Ayan N F. Discovery of Association Rules in Temporal Databases[C]//Proceedings of the International Conference on Information Technology. [s. l.]: Institute of Electrical and Electronics Engineers Computer Society, 2007: 371-376.
 - [2] 黄德才, 张良燕. 一种改进的关联规则增量式更新算法[J]. 计算机工程, 2008, 34(10): 38-42.
 - [3] 李志刚. 基于项集支持度的关联规则增量更新算法—BISUA[J]. 计算机工程与设计, 2007(9): 4072-4075.
 - [4] Tomas S, Bodagala S, Alsabti K, et al. An Efficient Algorithm for the Incremental Updation of Association rules in Large Databases[C]//In Proceedings of the International Conference on Knowledge Discovery and Data Mining. [s. l.]: Institute of Electrical and Electronics Engineers Computer Society, 1997: 263-266.
 - [5] 徐 勇, 周森鑫. 一种改进的关联规则挖掘方法研究[J]. 计算机技术与发展, 2006, 16(3): 77-79.
 - [6] 杨泽民. 一种改进的关联规则增量式更新算法[J]. 大同大学学报: 自然科学版, 2007(4): 112-124.
 - [7] Lee W J, Lee S J. An Efficient Mining Method for Incremental Updation in Large Databases[C]//4th International Conference on Intelligent Data Engineering and Automated learning. [s. l.]: Institute of Electrical and Electronics Engineers, 2003.
 - [8] 程玉胜, 邓小光, 江效尧. Apriori 算法中频繁项集挖掘实现研究[J]. 计算机技术与发展, 2006, 16(3): 58-60.
-
- V1.0.2[EB/OL]. 1999-11. <http://java.sun.com/jms>.
 - [3] Tai S, Rouvellou I. Strategies for integrating messaging and distributed object transactions[C]//IFIP/ACM. International conference on Distributed Systems Platforms and Open Distributed Processing. New York: Springer-Verlag, 2000: 308-330.
 - [4] Jalote P. Fault Tolerance in Distributed Systems[M]. USA: Prentice Hall, Inc, 1994.
 - [5] Shevat A. Distributed Enterprise Messaging with MantaRay[EB/OL]. 2004-12. <http://www.onjava.com/Pub/a/on-Java/2004/12/08/mantaRay.html>.
 - [6] 陈建巍, 李 京, 丁 柯. 可靠消息传输服务的设计和实现[C]//第七届计算机科学与技术研究生学术讨论会论文集. 中国计算机学会. 广元:[出版者不详], 2002.
 - [7] 王小霞, 陈 亮. 一种消息队列中间件的设计与实现[J]. 计算机工程, 2005, 31(21): 81-83.
 - [8] 赵 伟, 周 兵. 基于代理的消息中间件的设计与实现[J]. 计算机工程, 2004, 30(22): 91-92.
 - [9] 徐 晶, 许 玮. 消息中间件综述[J]. 计算机工程, 2005, 31(16): 73-76.