不确定数据频繁项集挖掘算法研究

赵学健[1],熊肖肖[1],张欣慧[2],孙知信[1]
(1. 南京邮电大学 现代邮政学院,江苏 南京 210003;
2. 南京邮电大学 物联网学院,江苏 南京 210003)

摘 要:频繁项集挖掘的目标是以频繁出现的项目集的形式发掘嵌入在海量数据中的隐式的、先前未知的、潜在的有用知识,以辅助决策。随着数据采集方式和传输方式的多样化,不确定数据在各种实际应用中大量出现。因此,近年来针对不确定数据的频繁项集挖掘算法的研究引起了学者的广泛关注。文中首先介绍了不确定数据的定义,并分析了不确定数据频繁项集挖掘的概率模型。接下来,将主流频繁项集挖掘算法分为 3 类:基于候选项集生成和测试的频繁项集挖掘算法,基于模式增长的频繁项集挖掘算法和基于生物启发的频繁项集挖掘算法,详细介绍了当前针对不确定数据的主流频繁项集挖掘算法,并对这些算法的性能进行了简单分析。最后,对不确定数据的频繁项集挖掘算法进行了总结与展望。

关键词:频繁项集;不确定数据;候选项集;模式增长;生物启发
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2019)07-0140-05
doi:10.3969/j.issn.1673-629X.2019.07.028

Research on Frequent Itemset Mining Algorithm for Uncertain Data

ZHAO Xue-jian[1], XIONG Xiao-xiao[1], ZHANG Xin-hui[2], SUN Zhi-xin[1]
(1. School of Modern Posts, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;
2. School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: The frequent itemset mining aims to explore the implicit, previously unknown and potential useful knowledge embedded in big data in the form of frequent itemsets to assist the decision-making. With the diversification of data acquisition mode and transmission mode, uncertain data appear in a large number of practical applications. Therefore, in recent years, the research on frequent itemset mining algorithm for uncertain data has attracted wide attention from scholars. In this paper, we first introduce the definition of uncertain data and analyze the probability model of mining frequent itemset for uncertain data. Then, we divide the typical frequent itemset mining algorithms into 3 categories: candidate generate-and-test based frequent itemset mining algorithms, pattern growth based frequent itemset mining algorithms and bio-inspired frequent itemset mining algorithms. Typical frequent itemset mining algorithms are introduced. Moreover, the performance of these algorithms is also analyzed. Finally, the algorithm of mining frequent itemsets for uncertain data is summarized and prospected.

Key words: frequent itemset; uncertain data; candidate itemset; pattern growth; bio-inspired

0 引 言

作为数据挖掘领域研究的重要分支之一,频繁项集挖掘的主要目的是以频繁出现的项目集的形式发掘嵌入在海量数据中的隐式的、先前未知的、潜在的有用知识[1-4]。当前,频繁项集挖掘在各领域应用广泛,如银行数据分析、市场营销、医疗诊断、气象数据分析等[5]。上述应用中广泛存在不确定数据,造成数据不确定性的原因主要有:对现实世界的有限感知和理解能力;感知监测设备的局限性;用于收集、储存、转换或数据分析的可用资源的限制;无线传输错误或网络延迟;数据粒度或隐私保护。因此,针对不确定数据的频繁项集挖掘引起了学者的广泛关注。

收稿日期:2018-08-08 **修回日期**:2018-12-11 **网络出版时间**:2019-03-21
基金项目:国家自然科学基金(61373135,61672299);国家自然青年科学基金(61702281,20140883);江苏省基础研究计划(自然科学基金)(BK20140883,BK20140894,BK20150869)
作者简介:赵学健(1982-),男,副教授,硕导,通信作者,CCF 会员(88401M),研究方向为无线传感器网络、大数据及物联网关键技术;熊肖肖(1995-),女,硕士研究生,研究方向为无线传感器网络、数据分析。
网络出版地址:http://kns.cnki.net/kcms/detail/61.1450.TP.20190321.0904.010.html

文中首先介绍了不确定数据的定义,并分析不确定数据频繁项集挖掘的概率模型。接着详细介绍了当前针对不确定数据的主流频繁项集挖掘算法,将主流频繁项集挖掘算法分为 3 类:基于候选项集生成和测试的频繁项集挖掘算法,基于模式增长的频繁项集挖掘算法和基于生物启发的频繁项集挖掘算法,分别介绍 3 类算法的代表性算法,并对相关算法的性能进行了简单分析。最后进行总结与展望。

1　定义及模型

在早期,最常见的频繁项集挖掘算法主要用于搜索传统的确定数据库。然而,在各种实际应用中,数据库中的每一条事务中项目的存在不是百分百确定的,而是依据某种相似性度量或是概率形式存在,比如说通过分析购物篮数据来预测商品需求量时,购物篮中的商品用户并不是肯定要购买的,而是存在一个不确定性。

在过去几年中,许多模型被提出用于不确定数据的分析[6-7],其中概率模型受到了众多研究者的青睐。从概率模型的角度来看,在不确定数据集 D 中,用户可能无法确定事务 T_q 中是否存在某个项目集 X,这种不确定性可以用存在概率 $p(X,T_q)$ 来表示,$p(X,T_q)$ 表示 X 存在于事务 T_q 中的可能性。存在概率 $p(X,T_q)$ 的取值最小为一个接近 0 的正值,表示 X 在 D 中存在的可能性不大,趋近于 0;存在概率 $p(X,T_q)$ 的取值最大为 1,表示项目集 X 绝对存在。从这个意义上说,传统的确定数据库可以看作每一个项目在事务 T_q 中的存在概率均为 100% 的不确定数据库。不确定数据的概率模型涉及以下三个定义。

定义 1(项集在事务中的存在概率):项集 X 在事务 T_q 中的出现概率记为 $p(X,T_q)$,等于项集中各项目在事务 T_q 中存在概率的乘积。

即:

$$p(X,T_q)=\prod_{I_j\in X}p(I_j,T_q),$$

其中,I_j 为项集 X 的第 j 个项目。

定义 2(项集的期望支持度):项集 X 在不确定数据库 D 中的期望支持度记为 $\exp\mathrm{Sup}(X)$,等于项集 X 在包含该项集的所有事务中的出现概率之和。

即:

$$\exp\mathrm{Sup}(X)=\sum_{X\subseteq T_q\wedge T_q\in D}p(X,T_q)=\sum_{X\subseteq T_q\wedge T_q\in D}\Big(\prod_{I_j\in X}p(I_j,T_q)\Big).$$

定义 3(频繁项集):在不确定数据库 D 中,当项集 X 的期望支持度大于等于最小期望支持度时(最小期望支持度为最小期望支持度阈值 δ 与数据库 D 中所包含事务数的乘积),则项集 X 为频繁项集,即当 $\exp\mathrm{Sup}(X)\geqslant\delta\times|D|$ 时,项集 X 为频繁项集。

2　频繁项集挖掘算法

2.1　基于候选项集生成和测试的频繁项集挖掘算法

从不确定数据中挖掘频繁项集的一种重要方法是首先生成候选频繁项集然后扫描数据库对候选频繁项集进行测试。基于这个思想,Chui 等[8]提出了 U-Apriori 算法,该算法从不确定数据中挖掘频繁模式的方法是分层的、宽度优先的、自下而上的。具体来说,U-Apriori 算法首先计算所有 1-项集的期望支持度,通过扫描不确定数据库,那些期望支持度大于最小期望支持度的项目最终组成频繁 1 项集。然后,U-Apriori 算法反复应用候选项集生成和测试的过程,从频繁 k 项集生成候选 $k+1$ 项集,并测试它们是否是频繁的 $k+1$ 项集。U-Apriori 算法与用于挖掘确定数据的 Apriori 算法[2]是一样的,都依赖于 Apriori 属性,即频繁模式的所有子集也必须是频繁的,反之任何非频繁项集的所有超集也是非频繁项集,该性质也称为向下闭包特性。

U-Apriori 算法还通过采用 LGS-修整策略来提高其效率,LGS-修整包括局部修整、全局剪枝和单次修补。

该策略从不确定数据的原始概率数据集 D 中剔除每一项存在概率低于用户指定的修剪阈值的项,然后从修剪以后的数据集中挖掘频繁项集。

Chui 等[9]采用递减修剪技术,进一步提高了 U-Apriori 的效率。递减修剪技术通过估计候选频繁项集的期望支持度的上限以减少候选模式的数量。如果候选频繁项集 X 的估计期望支持度上限值低于最小期望支持度,则立即剪除该候选频繁项集 X。

2.2　基于模式增长的频繁项集挖掘算法

基于模式增长的频繁项集挖掘算法是基于候选项集生成和测试的频繁项集挖掘方法的一种有效替代方法,能够避免产生大量的候选项集。常用的模式增长频繁项集挖掘算法又可以分为两类:基于超链接结构的频繁项集挖掘算法和基于树结构的频繁项集挖掘算法。

(1)基于超链接结构的频繁项集挖掘算法。

基于超链接结构的频繁项集挖掘算法以超链接结构存储数据集的内容,在这类算法中,频繁的模式以深度优先、分而治之的方式被挖掘出来。

Aggarwal 等[10]提出了一种基于超链接结构的 UH-mine 算法,用于从不确定数据中挖掘频繁项集。该算法用一个名为 UH-struct 的超链接结构存储不确定数据集 D 中的概率数据内容。UH-struct 中的每一行表

示不确定数据集 D 中的一个事务,与用于挖掘确定性数据的 H-struct 不同的是,UH-struct 还存储了项目的存在概率。换句话说,对于存在于事务中的每个项目,通过 UH-struct 可以得到:项目本身;项目的存在概率;项目的超链接结构。因此,在 UH-mine 算法中,通过建立的 UH-struct 可以递归地扩展每个频繁项集并调整其在 UH-struct 中的超链接来实现频繁项集的挖掘。

该算法在较小的数据集上可以获得较好的效果,然而在大数据集上,由于 H-struct 需要较多的空间来存储数据集,并且算法需要多次递归调用,因此时间效率并不理想。

(2)基于树结构的频繁项集挖掘算法。

基于候选项集生成和测试的频繁项集挖掘算法使用了分层的自下而上的广度优先挖掘范式,但是往往生成的候选项集数量过多。基于超级链接结构的频繁项集挖掘算法通过递归地调整超链接结构,以深度优先的方式从不确定的数据中找到频繁的模式,时间效率也不理想。

基于树的频繁项集挖掘算法使用树结构对不确定数据集进行存储,并采用深度优先、分而治之的方法挖掘频繁项集,既可避免产生许多候选项集,又避免了递归地调整多个超链接结构,具有较好的性能。

Leung 等提出了一种基于树的频繁项集挖掘算法 UF-Growth[11],类似于用于挖掘确定性数据的频繁项集挖掘算法 FP-Growth[12],UF-Growth 也是通过构造一个树结构来存储数据集的内容。但是,它不使用 FP-tree,因为 FP-tree 中的每个节点都只能记录项目及其在树路径中的出现数目。当挖掘确定性数据集时,项集 X 的期望支持度取决于项集 X 中各项目的出现次数。然而,在挖掘不确定数据时,X 的期望支持度是 X 中各项目的发生计数和存在概率的乘积之和。因此,UF-Growth 算法采用了不同于 FP-tree 的树结构 UF-tree。

UF-tree 中的每个节点由三个部分组成:项目、存在概率、项目在路径中的出现次数。UF-tree 的构造方式与 FP-tree 的构造类似,但是只有当事务和子节点中存在相同的项和相同的存在概率时,新事务才会与子节点合并。因此,UF-tree 比原来的 FP-tree 具有更低的压缩比。

Aggarwal 等[13]提出了 UFP-growth 算法,与 UF-growth 算法一样,UFP-growth 算法也会通过两次扫描不确定数据集来建立 UFP-tree。UFP-tree 中,当项集 X 对应的节点具有相似的存在概率值时,会聚集成一个超级节点。超级节点将会存储项集 X、存在概率值及其出现数信息。UFP-growth 算法在第二次扫描不确定数据集时才能发现所有真正频繁的模式,然而由于 UFP-growth 的近似性质,UFP-growth 算法除了能够发现那些真正频繁的项集之外,还会发现一些不常见的模式,称为假阳性模式。

因此,需要对不确定数据集进行第三次扫描,以消除这些错误。

Leung 和 Tanbeer[14]提出了一种用于不确定数据集的频繁项集挖掘算法,称为 CUF-growth 算法。该算法构建了一种叫做 CUF-tree 的树结构,与 UFP-growth 算法一样,CUF-growth 算法也对不确定数据集进行三次扫描,以挖掘频繁项集。CUF-growth 首先扫描数据集以计算事务上限,然后通过第二次扫描数据集构建 CUF-tree。在第二次扫描不确定数据集结束时,CUF-growth 算法会发现所有潜在的频繁项集。由于这些潜在的频繁项集包括所有真正频繁的项集和一些并不频繁的项集,CUF-growth 算法最后通过第三次快速扫描数据集,以检查每个项目集,验证它们是否是真正频繁的,即修剪假阳性项目集。

Leung 和 Tanbeer[15]引入了前缀项上限的概念,并提出了相应的 PUF-growth 算法用于挖掘不确定数据集中的频繁项集。与 UFP-growth 和 CUF-growth 一样,PUF-growth 算法也对不确定数据的概率数据集进行三次扫描,以挖掘频繁项集。在第一次扫描中,PUF-growth 计算前缀项上限。在第二次扫描中,PUF-growth 构建 PUF-tree。PUF-growth 算法也是在第二次扫描不确定数据集结束时找到所有潜在频繁项集。因为这些潜在的频繁项集包括所有真正频繁的项集和一些罕见的项集。PUF-growth 算法最后通过第三次快速扫描数据集,检查每个数据集是否真的频繁。PUF-growth 算法采用的前缀项上限与 CUF-tree 算法的事务上限相比更加接近项集的期望支持度,因此在第三次扫描过程中,需要检验的假阳性项目集数量通常小于 CUF-growth 算法需要检验的项目集数量,速度更快。

2.3 基于生物启发的频繁项集挖掘算法

上述基于候选项集生成与测试的频繁项集挖掘算法以及基于模式增长的频繁项集挖掘算法都是精确挖掘算法,也就是说这些算法都可以挖掘出数据集中的所有频繁项集。这类算法虽然精确度高,但是时间复杂度通常与数据集的规模成正比。当数据集非常大时,比如社交网络数据[16]或者大型生物信息数据集[17],精确的频繁项集挖掘算法几乎是无能为力的[18]。

为了解决精确频繁项集挖掘算法时间效率低下的问题,研究人员提出了基于生物启发的频繁项集挖掘算法,比如基于遗传算法的频繁项集挖掘算法[19-20],

基于群体智能的频繁项集挖掘算法等[21]。基于生物启发的频繁项集挖掘算法可以在规定的合理时限内完成,但是通常该类算法不能挖掘出所有的频繁项集,称之为模糊频繁项集挖掘算法。因此,针对大型的不确定数据集,提出合适的生物启发频繁项集挖掘算法在保证时间效率的前提下获得更高质量的频繁项集,将会是一个持续的挑战。

文献[19]提出的 GA-FIM 算法将每一个项目集看成由 n 个元素组成的向量,如果该项目集的第 i 个元素属于该项目集,则将向量的第 i 个元素设置为 1,否则设置为 0。比如说数据集中包含 5 个项目,即 $I=\{a,b,c,d,e\}$,则项目集 bde 可以用向量 $\{0,1,0,1,1\}$ 表示。接下来,在完成初始化后将执行遗传算法的交叉、变异和选择操作,直到达到预先设置的轮数为止。最终,GA-FIM 算法得到的频繁项集为每一轮过程中发现的频繁项集的集合。

文献[21]最早提出了 PSO-FIM 算法,文献[22]后期又对 PSO-FIM 算法进行了改进。在 PSO-FIM 算法中,群体中的每一个粒子代表一个项目集,粒子在初始化时随机设置为项目集空间中的任意一个项目集。PSO-FIM 算法初始化时首先构造一群随机粒子,接下来通过迭代找到最优解。在每一次迭代中,粒子通过跟踪两个极值来进行更新,第一个值是本粒子找到的最优解,即具有最大期望支持度的项目集,第二个值是整个粒子群找到的最优解。

文献[19]和文献[21]表明,GA-FIM 算法和 PSO-FIM 算法相对于当前的精确频繁项集挖掘算法具有更好的时间效率。然而,这两种算法所得到的频繁项目集结果并不理想,也就是说算法的最终输出仅为频繁项目集的一个子集。

3　算法性能对比分析

上述对不确定数据集的频繁项目集挖掘算法进行了分类划分,并对代表性算法进行了介绍,包括 U-Apriori、UH-mine、UF-growth、UFP-growth、CUF-growth、PUF-growth、GA-FIM 及 PSO-FIM 等。

在准确性方面,U-Apriori、UH-mine、UF-growth、UFP-growth、CUF-growth、PUF-growth 算法都可以返回满足用户指定最小期望支持度阈值的所有频繁项集。然而,GA-FIM 及 PSO-FIM 算法由于融合了启发式智能算法的思想,不一定能够返回所有符合条件的频繁项目集,准确性相对较差。

在内存消耗方面,U-Apriori 保留了候选频繁项集列表,而基于树和超链接结构的算法则构造内存结构,比如 UF-tree 及其变体,扩展的 H-struct 等。一方面,UF-tree 比扩展的 H-struct 更紧凑,即需要更少的空间。然而,另一方面,UH-mine 只保留一个扩展的 H-struct,而基于树的算法通常构造不止一棵树,树的大小也可能不同。

因此,基于树和超链接结构的算法的内存消耗要根据具体算法和数据集情况来确定。GA-FIM 及 PSO-FIM 算法不需要保存大量候选频繁项集或其他内存结构,内存消耗较小。

就时间性能而言,GA-FIM 及 PSO-FIM 等基于生物启发的频繁项集挖掘算法可以灵活设置,通常优于其他几种算法。

基于候选项集生成和测试及基于模式增长的频繁项集挖掘算法的时间性能影响因素较多。首先,项目对应的存在概率会影响算法的时间性能。通常不确定数据集中的项目呈现较低的存在概率时,大多数算法的性能都很好,因为这些数据集不会导致较长的频繁模式。当项目呈现较高的存在概率值时,U-Apriori 算法会生成更多的候选频繁项集,U-growth 算法需要构造更多更大的 UF-tree,UH-mine 算法需要调整更多的超链接结构,自然都需要更长的运行时间。其次,当最小期望支持度减小时,往往会得到更多的频繁项目集,当然也需要更长的运行时间。此外,数据集的密度也会影响运行时间。例如,当数据集密集时,UF-tree 获得了更高的压缩比,因此与稀疏数据集相比,遍历所需的时间更短,时间性能更好。

4　结束语

频繁项集挖掘是一项重要的数据挖掘任务,有助于发现隐式的、先前未知的潜在有用的知识,有助于揭示许多现实生活应用中共同发生的项目。由于许多现实生活应用中的数据都具有不确定性,因此,近年来不确定数据的频繁项集成为研究者们关注的焦点。文中介绍了不确定数据频繁项集挖掘算法的相关成果,其中包括基于候选项集生成和测试的、基于模式增长的以及基于生物启发的频繁项集挖掘算法,并对典型代表性算法进行了介绍,对其性能进行了分析。

未来的可能研究方向包括:(1)基于生物启发的频繁项集挖掘算法的改进创新;(2)在社交网络分析等应用领域的不确定数据中挖掘频繁项集;(3)从不确定数据中挖掘频繁序列和频繁图;(4)不确定频繁模式的可视化分析。

参考文献:

[1] 王　乐,常艳芬,王　水. 基于模式增长的不确定数据的频繁模式挖掘算法[J]. 计算机应用,2015,35(7):1921-1926.

[2] AGGRAWAL R, SRIKANT R. Fast algorithms for mining

association rules in large databases[C]//International conference on very large data bases. [s. l.]: Morgan Kaufmann Publishers Inc., 1994: 487-499.

[3] 汪金苗,张龙波,邓齐志,等. 不确定数据频繁项集挖掘方法综述[J]. 计算机工程与应用,2011,47(20):121-125.

[4] 王楠楠,刘慧婷. 频繁模式挖掘系统的设计与开发[J]. 计算机技术与发展,2018,28(2):150-153.

[5] BHOGADHI V,CHANDAK M B. A review of frequent pattern mining algorithms for uncertain data[C]//Proceedings of SAI intelligent systems conference. [s. l.]: Springer, 2018: 974-983.

[6] AGGARWAL C C. Managing and mining uncertain data [M]. US: Springer, 2009.

[7] AGGARWAL C C, YU P S. A survey of uncertain data algorithms and applications[J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(5): 609-623.

[8] CHUI C K, KAO B, HUNG E. Mining frequent itemsets from uncertain data[C]//Pacific-Asia conference on advances in knowledge discovery and data mining. [s. l.]: Springer-Verlag, 2007: 47-58.

[9] CHUI C K, KAO B. A decremental approach for mining frequent itemsets from uncertain data[C]//Proceedings of the 12th Pacific-Asia conference on advances in knowledge discovery and data mining. Osaka, Japan: Springer-Verlag, 2008: 64-75.

[10] TONG Yongxin, CHEN Lei, CHENG Yurong, et al. Mining frequent itemsets over uncertain databases[J]. Proceedings of the VLDB Endowment, 2012, 5(11): 1650-1661.

[11] LEUNG K S, MATEO M A F, BRAJCZUK D A. A tree-based approach for frequent pattern mining from uncertain data[C]//Proceedings of the 12th Pacific-Asia conference on advances in knowledge discovery and data mining. Osaka, Japan: Springer-Verlag, 2008: 653-661.

[12] HAN Jiawei, PEI Jian, YIN Yiwen. Mining frequent patterns without candidate generation[J]. ACM SIGMOD Record, 2000, 29(2): 1-12.

[13] AGGARWAL C C, LI Yan, WANG Jing. Frequent pattern mining with uncertain data[C]//Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining. Paris, France: ACM, 2009: 29-38.

[14] LEUNG K S, TANBEER S K. Fast tree-based mining of frequent itemsets from uncertain data[C]//Proceedings of the 17th international conference on database systems for advanced applications. Busan, South Korea: Springer-Verlag, 2012: 272-287.

[15] LEUNG K S, TANBEER S K. PUF-tree: a compact tree structure for frequent pattern mining of uncertain data[C]//Pacific-Asia conference on knowledge discovery and data mining. [s. l.]: [s. n.], 2013: 13-25.

[16] KOÇAK Y, ÖZYER T, ALHAJJ R. Utilizing maximal frequent itemsets and social network analysis for HIV data analysis[J]. Journal of Cheminformatics, 2016, 8(1): 71-87.

[17] SOHRABI M K, ROSHANI R. Frequent itemset mining using cellular learning automata[J]. Computers in Human Behavior, 2017, 68: 244-253.

[18] DJENOURI Y, BENDJOUDI A, MEHDI M, et al. GPU-based bees swarm optimization for association rules mining [J]. Journal of Supercomputing, 2015, 71(4): 1318-1344.

[19] DJENOURI Y, BENDJOUDI A, NOUALI-TABOUDJEMAT N. Association rules mining using evolutionary algorithms[C]//9th international conference on bio-inspired computing: theories and applications. Wuhan, China: [s. n.], 2014.

[20] MARTÍN D, ALCALÁ-FDEZ J, ROSETE A, et al. NICGAR: a niching genetic algorithm to mine a diverse set of interesting quantitative association rules[J]. Information Sciences, 2016, 355-356: 208-228.

[21] KUO R J, CHAO C M, CHIU Y T. Application of particle swarm optimization to association rule mining[J]. Applied Soft Computing, 2011, 11(1): 326-336.

[22] LIN C W, YANG L, FOURNIER-VIGER P, et al. Mining high-utility itemsets based on particle swarm optimization [J]. Engineering Applications of Artificial Intelligence, 2016, 55: 320-330.