广西科学院学报
Journal of Guangxi Academy of Sciences

2018,34(2):137～142,150
Vol. 34,No. 2 May 2018

DOI:10. 13657/j. cnki. gxkxyxb. 20180604. 001

戴瑀君,徐周波. 基于 SAT 和 BDD 的频繁序列挖掘技术[J]. 广西科学院学报,2018,34(2):137-142,150.

DAI Y J,XU Z B. A boolean satisfiability and binary decision diagram based approach for mining frequent sequence[J]. Journal of Guangxi Academy of Sciences,2018,34(2):137-142,150.

基于 SAT 和 BDD 的频繁序列挖掘技术*

Frequent Sequence Mining Techniques based on SAT and BDD

戴瑀君,徐周波

DAI Yujun,XU Zhoubo

(桂林电子科技大学计算机与信息安全学院,广西桂林 541004)

(School of Computer and Information Security, Guilin University of Electronic Technology, Guilin,Guangxi,541004,China)

摘要:【目的】研究模式挖掘领域中的频繁序列挖掘技术,由于序列模式挖掘存在指数级的搜索空间,且传统的 SAT 求解算法无法高效求解大规模数据集的缺点,因此研究符号表示和操作技术,用来避免冗余计算。**【方法】**提出基于 SAT 的频繁序列挖掘的符号 OBDD 算法,基于深度优先算法的思想,首先将频繁序列挖掘问题构建为 SAT 模型,其次对变量进行排序并将约束子句分类后分别描述为 OBDD,利用 OBDD 的"与"操作得到满足 SAT 的所有频繁序列模式。**【结果】**实例结果表明,该方法准确可行。**【结论】**该方法能有效缩减搜索空间,提高求解效率。

关键词:布尔可满足性 有序二叉决策图 频繁序列挖掘

中图分类号:TP301. 6 **文献标识码:**A **文章编号:**1002-7378(2018)02-0137-06

Abstract:【Objective】To research the frequent sequence mining techniques in the field of pattern mining. The sequential pattern mining problem has an exponential search space. However, the traditional SAT solver algorithm cannot efficiently solve large-scale data sets. For this shortcoming, the symbolic representation and operation techniques are studied to avoid redundant computing.**【Methods】**Based on the depth-first algorithm, the symbolic OBDD algorithm based on SAT for mining frequent sequence was proposed. Firstly, the frequent sequence mining problem was constructed as a SAT model. Second, the variables were sorted and the constraint clauses were classified and described as OBDD respectively. Then the "AND" operation of OBDD was used to find all the frequent patterns that satisfied the SAT.**【Results】**The example results showed that this method was accurate and feasible.**【Conclusion】**This approach could reduce the search space effectively and improve the efficiency.

Key words: boolean satisfiability, ordered binary decision diagram, frequent sequence pattern mining

收稿日期:2018-01-10

作者简介:戴瑀君(1992—),女,硕士研究生,主要从事符号计算、约束求解和模式挖掘研究,E-mail:daiyujun9123@outlook.com。

*广西自然科学基金项目(2017GXNSFAA198172)资助。

0 引言

【研究意义】频繁序列是一种基本的序列模式,定义为相对于给定的最低频率阈值,频繁发生的事件或项目的有序列表,其中项目可以是 DNA 序列

中的核苷酸。序列模式挖掘是数据挖掘中的一项重要任务,在模式挖掘中,频繁序列挖掘问题已被深入研究,是计算生物学、时间序列分析和文本挖掘的核心任务,应用领域广泛,包括分析时间戳购物数据,挖掘网络访问模式(WAP)[1],查找DNA序列中的相关基因[2],以及分类蛋白质[3]等。在实际中,序列模式挖掘被广泛地应用于各种序列数据集中,如生物信息学上的基因微阵列数据,从中挖掘哪些基因组合模式在某类病人中会频繁出现;以单词作为项目(item)的文档序列,研究在不同文档中单词序列的出现模式;用户点击流数据,用于挖掘用户的频繁点击模式,建立用户模型,完善网站功能与UI结构。除此之外,只要是序列数据集,都可以利用序列模式挖掘获得规律。**【前人研究进展】**目前序列挖掘的研究主要分为两类。一类将基于约束规划(CP)技术与模式挖掘相结合。2008年,De Raedt等[4]针对项集挖掘(CP4IM)提出CP的数据挖掘(DM)框架,提供了一个表述性的且灵活的表示模型。模式的新的约束通常需要在特定的方法中实现,但可以轻松地集成到该CP框架中,使数据挖掘问题受益于几种通用和高效的CP求解技术。2011年Guns等[5]提出在项集挖掘中使用的一些典型约束(例如频率,极大性,单调性)如何在CP中制定使用。这项研究是项集挖掘的第一个CP方法,显示出良好的表述性因素。2013年,Jabbour等[6]提出一种基于布尔可满足性问题(SAT)的编码,用于发现项目序列以及项集序列中的频繁、闭合和最大模式的问题。这种新的研究趋势为人工智能(AI)和数据挖掘之间的交叉渗透提供了很好的机会。另一类,对当前最常用且最有效的序列挖掘技术——模式增长(pattern growth)[7]方法进行改进。由于频繁序列挖掘器需要探索指数级的搜索空间,使得在大量长频繁序列模式存在的情况下,如在具有小字母表的DNA或蛋白质序列数据集中,或者在支持度阈值较低时、字母表较大的weblog数据集中,挖掘极具挑战性。因此有研究者提出基于二叉决策图(BDD)的数据挖掘方法:2009年,Loekito等[8]提出了一种称为SeqBDD的新的BDD类型,并成功将其应用于频繁子序列挖掘,展现出BDD的良好压缩性能。2010年,Cambazard[9]通过将所有项集的集合编译成BDD来解决频繁项集挖掘问题,然后通过查询BDD来提取频繁的项集。**【本研究切入点】**鉴于CP技术的表述性和灵活性,但存在搜索空间爆炸的问题;而二叉决策图(BDD)及其拓展结构具有高紧凑性和易操作性,一定程度上可以解决搜索空间爆炸问题,为处理大规模数据提供技术支撑。因此,提出基于SAT的频繁序列挖掘的符号OBDD算法,解决模式挖掘中输出尺寸巨大,很难快速从中检索相关信息等问题。**【拟解决的关键问题】**将频繁序列挖掘问题构建为SAT模型,通过对SAT的符号OBDD表示,并通过对SAT中约束进行OBDD操作给出了基于SAT的频繁模式挖掘的符号求解技术。最后通过实例证实算法的可行性以及准确性。

1 预备知识

1.1 布尔可满足性问题(SAT)

布尔可满足性问题是一个判定经典命题逻辑公式是否一致的问题,是研究最多的多项式复杂程度的非确定性问题(NP-C问题)之一。

定义1.1.1 合取范式Φ(Conjunctive Normal Form,简称CNF):命题变元(p)及其否定($\neg p$)称为文字(literals);有穷个文字的析取构成子句(clause);有穷个子句的合取(conjunction)组成合取范式Φ。

合取范式Φ可以理解为一些布尔变量的或与表达式,表达式中只能使用与、或、非。使用线性Tseitin的编码[10]可以将任何命题公式转化为CNF。以下几种都是合取范式:

$\neg A \wedge (B \vee C)$

$(A \wedge B) \wedge (\neg B \vee C \vee \neg D) \wedge (D \vee \neg E)$

$(\neg A \vee B)$

$A \wedge B$

定义1.1.2 解释(interpretation)M:给定一个合取范式Φ以及Φ的一个赋值M,M为一个模型,当且仅当$M(\Phi)=$true,称M满足合取范式Φ。

定义1.1.3 布尔可满足问题:给定一个布尔公式Φ(本文布尔公式以合取范式表示),若存在Φ的一个解释M,使得解释M满足布尔公式Φ,称布尔公式Φ是可满足的(satisfiable);如果在Φ的赋值空间内不存在这样的解释M,则称布尔公式Φ是不可满足的(unsatisfiable)。

1.2 SAT求解器

目前大多数高效的SAT求解器都是基于DPLL算法[11](Davis-Putnam-Logemann-Loveland),它是一种完全的、基于回溯的算法。DPLL算法主要采用了深度优先搜索(Depth-first Search,DFS)算法思想。深度优先搜索算法指的是按照某种条件往前试探搜索,如果前进中遭到失败

则回溯到上一层结点另选通路继续搜索,直到找到符合条件的目标为止。基本的 DPLL 算法包含 3 个主要步骤:

①变量决策:在搜索树的每个层级上选择一个还未被赋值的决策变量并赋为一个布尔值。

②赋值过程:推导并传播强制文字分配,简化子句,减少原始问题中的子句数。

③回溯过程:当搜索遇到冲突时,搜索过程从深的决策层返回到浅决策层。冲突子句(学习子句)通过自下而上遍历蕴涵图(implication graph)产生[12-13]。当产生的冲突子句仅包含一个来自当前决策层的文字时,学习或冲突分析过程停止。该冲突子句表明,当前层级的唯一文字(称为断言文字)隐含在上一层级(称为断言层),是该子句中其他文字的最大层级。求解器回溯到断言层并将断言文字分配为 true。当生成空冲突子句时,原始公式记录为不可满足;当找到模型时,该公式为可满足的,记录模型。

1.3 频繁序列挖掘(FPS)

定义 1.3.1 项目序列(Sequences of items):给定项目的有限集合字母表 Σ,通配符是不在 Σ 中的新符号,该符号可匹配字母表中的任一符号。Σ 上的项目序列 s 由属于 Σ 的符号 $s_0\cdots s_{n-1}$ 表示。其长度表示为 $|s|$,序列中符号的所有位置由集合 $P_s=\{0,\cdots,|s|-1\}$ 表示。

定义 1.3.2 模式(Pattern):Σ 上的模式为序列 $x=x_0\cdots x_{m-1}$,其中 $x_0\in\Sigma$,$x_{m-1}\in\Sigma$,$x_i\in\Sigma\cup\{*\}$,$i=1,\cdots,m-2$。

若 $\forall i\in\{0,\cdots,m-1\}$,$x_i=s_{l+i}$ 或 $x_i=*$,则称在序列 s 的位置 $l\in P_s$ 处,x 包含于序列 $s=s_0\cdots s_{n-1}$,记为 $x\leqslant_l s$。如果对于 $\exists l\in P_s$ 使得 $x\leqslant_l s$,则称 x 包含于 s。x 覆盖于 s 定义为集合 $L_s(x)=\{l\in P_s\mid x\leqslant_l s\}$。$x$ 在 s 中的支持度的值为 $|L_s(x)|$。

定义 1.3.3 频繁模式:给定序列 s,模式 x,最小支持度阈值 $\lambda\geqslant 1$,若 $L_s(x)\geqslant\lambda$,则 x 是序列 s 中相对于支持度 λ 频繁的模式。一个项目序列中的频繁模式挖掘问题包含计算所有相对于 λ 频繁的模式集合 M_s^{λ}。

例如,对于序列 $s=aaccbcabcba$,模式 $x=a*c$。由于 $x\leqslant_0 s$,$x\leqslant_1 s$,$x\leqslant_6 s$,得到 $L_s(x)=\{0,1,6\}$,这种情况下,若假定最小支持度阈值等于 3,那么模式 x 是序列 s 的一个频繁模式。

1.4 二叉决策图(BDD)

BDD[14]是一个规范的有向无环图(DAG)数据结构,用于紧凑地表示布尔公式。BDD 也可以视为相同子树合并的二叉决策树。有序二叉决策图(OBDD)[15]是布尔函数的一种有效图形和数学描述技术。

定义 1.4[16] 对于从 $\{0,1\}^n$ 到 $\{0,1\}$ 的布尔函数 $f(x_1,x_2,\cdots,x_n)$ 和给定的变量序 π,一个 OBDD 就是用于表示布尔函数 $f(x_1,x_2,\cdots,x_n)$ 的一个有向无环图,它满足:

①OBDD 中的结点分为非终结点和终结点两类。没有后继子结点或者没有输出弧的结点称为终结点,即终结点 0 和终结点 1,分别表示布尔常量 0 和 1;除终结点之外的结点称为非终结点;

②每个非终结点 u 具有四元组属性($pointer(u),var(u),low(u),high(u)$),其中,$pointer(u)$ 表示结点 u 所对应的布尔函数;$var(u)$ 表示结点 u 的标记变量;$low(u)$ 表示结点 u 所对应的函数中变量 $var(u)$ 取值为 0 时对应的 0-分支子结点;$high(u)$ 表示结点 u 所对应的函数中变量 $var(u)$ 取值为 1 时对应的 1-分支子结点;

③每个非终结点有且仅有两条输出弧,将它们和各自的两个分支子结点连接在一起。结点 u 和 $low(u)$ 的连接弧称为 0-边,结点 u 和 $high(u)$ 的连接弧称为 1-边;

④对于 OBDD 中的任意结点 u,都有 $low(u)\neq high(u)$;对于 OBDD 中的任意两个 $var(u)=var(v)$ 的不同结点 u 和 v,都有 $low(u)\neq low(v)$,或者 $high(u)\neq high(v)$,或者 $low(u)\neq low(v)$ 且 $high(u)\neq high(v)$;

⑤在 OBDD 的任一有向路径上,布尔函数 $f(x_1,x_2,\cdots,x_n)$ 中的每个变量均以变量序 π 所规定的次序依次最多出现一次。

定义 1.4 中所定义的 OBDD 实质上就是简化的 OBDD,即 ROBDD(Reduced OBDD)。

例如,对于布尔函数 $f=(x_1+x_2)\cdot x_3$ 在变量序:$x_1<x_2<x_3$ 下所对应的 OBDD 如图 1 所示。在布尔函数的 OBDD 的表示中,对于变量的一组赋值,所得到的函数值由根结点到一个终结点的一条路径决定。这条路径所对应的分支由变量的这组赋值来决定,该分支的终结点所标识的值就是变量在这组赋值下所对应的函数值。

在 OBDD 中提供了以下 2 条简化规则。

规则 1(删除规则):对于 OBDD 中的结点 u,如果 $low(u)=high(u)$,则删除结点 u,并将结点 u 的所有入边指向 $low(u)$ 结点;

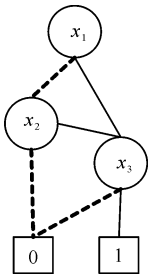
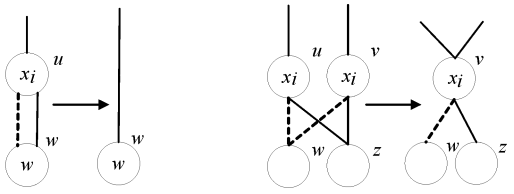


图 1 布尔函数 $f=(x_1+x_2)\cdot x_3$ 的 OBDD 表示

Fig. 1 OBDD for Boolean function $f=(x_1+x_2)\cdot x_3$

规则 2(合并规则):对于 OBDD 中的结点 u 和 v,如果 $var(u)=var(v)$,$low(u)=low(v)$ 且 $high(u)=high(v)$,则删除其中一个结点,并将被删除结点的所有入边指向保留结点。该规则同样适用于具有相同标号的终结点。

OBDD 中的删除规则和合并规则如图 2 所示。



(a)删除规则 Delete rule (b)合并规则 Merge rule

图 2 OBDD 的简化规则

Fig. 2 Simplified rules for OBDD

2 FPS 问题的 SAT 符号模型构建

2.1 FPS 问题的 SAT 编码

基本思想在于使用命题变量来表示候选模式中元素的位置,以及使用基数约束[6]推导候选模式的支持度。

令 $\Sigma=\{e_1,\cdots,e_m\}$ 为字母表,s 为 Σ 上的序列,长度为 n,λ 为最小支持度阈值。将 s 中出现的每个字符 e 关联到 k_e 个命题变量的集合 $x_{e,0},\cdots,x_{e,(k_e-1)}$ 使得 $k_e=\min(\max(L_s(e))+1,n-\lambda+1)$,变量 $x_{e,i}$ 表示 e 在候选模式中,位置为 i。$x_{e,i}=1$ 表示候选模式中存在元素 e,且位于模式的第 i 位,$x_{e,i}=0$ 表示元素 e 不位于模式的第 i 位。$\{0,\cdots,\min(\max(L_s(e)),n-\lambda)\}$ 对应于在候选模式中 e 的所有可能位置集合,因此只需将 $\min(\max(L_s(e))+1,n-\lambda+1)$ 个变量关联到每个字符 e,减少约束个数,以便加速求解。

约束 1:第一个符号必须是一个固定的字符(不同于通配符)。该属性由以下简单子句表示:

$$\bigvee_{e\in\Sigma}x_{e,0};$$

约束 2:由二进制子句组成的以下约束获取候选模式不存在的位置:

$$\bigvee_{e\in\Sigma,0\leqslant l\leqslant n-1,0\leqslant i\leqslant k_e-1}(x_{e,i}\wedge s_{l+i}\neq e)\rightarrow y_l,$$

其中 $y_0,\cdots,y_{n-1}$ 是 n 个新的命题变量。如果候选模式不在 s 中的位置 l,则在上述公式中 $y_l=1$。在经典命题逻辑中,$A\rightarrow B:=\neg A\vee B$,因此上述公式可以看作是二进制子句集(表达式 $s_{l+i}\neq e$ 是常数,即 $s_{l+i}\neq e\in\{0,1\}$)。

约束 3:在枚举序列 s 中相对于支持度阈值 λ 的所有频繁模式问题中,需要表示候选模式至少出现 λ 次。该属性通过以下基数约束获得:

$$\sum_{l=0}^{n-1}y_l\leqslant n-\lambda。$$

若该约束不满足,则意味着至少存在 $n-\lambda+1$ 个位置候选模式不出现。可见,候选模式出现的位置最多存在 $\lambda-1$ 个,即不是频繁的。否则至少存在 λ 个候选模式的位置,即该模式为频繁的。因此,该约束能够推导出所考虑候选模式的支持度,以判定它是否大于或等于最小支持度阈值。

在给定序列 s 中枚举所有频繁模式的问题由约束 1、约束 2 和约束 3 表示。

例:考虑序列 $aabb$ 的频繁模式挖掘问题,最小支持度阈值为 2。编码对应于以下公式:

约束 1:$x_{a,o}\vee x_{b,o}$, (1)

约束 2:$x_{a,o}\rightarrow(y_2\wedge y_3)$, (2)

$x_{a,1}\rightarrow(y_1\wedge y_2\wedge y_3)$, (3)

$x_{b,o}\rightarrow(y_0\wedge y_1)$, (4)

$x_{b,1}\rightarrow(y_0\wedge y_3)$, (5)

$x_{b,2}\rightarrow(y_2\wedge y_3)$, (6)

约束 3:$y_0+y_1+y_2+y_3\leqslant 2$。 (7)

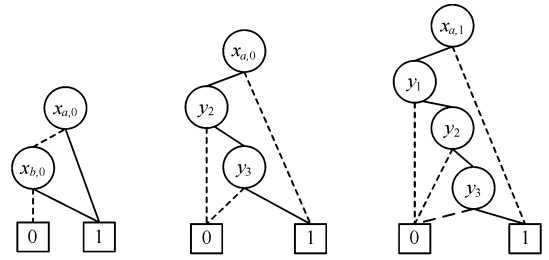
对于所有的布尔解释 M,若 $M(x_{a,i})=M(x_{b,i})$,则 $y_0+y_1+y_2+y_3=4$。因此,$M(x_{a,i})\neq M(x_{b,i})$,表示在同一位置不能有不同的实体字符。此外,由约束 3 可得对于所有布尔解释 M,必须有 $M(x_{a,1})=0$。即元素 a 不出现于所有候选模式的第 1 位。用 $\{x_{a,0},x_{a,1},x_{a,2},x_{b,0},x_{b,1},x_{b,2}\}$ 的子集描述公式的每个布尔模型,可得模式为 $\{\{x_{a,o}\},\{x_{b,o}\}$ 和 $\{x_{a,o},x_{b,2}\}\}$。对应于模式 a,b 和 $a*b$。

2.2 SAT 的符号 OBDD 表示

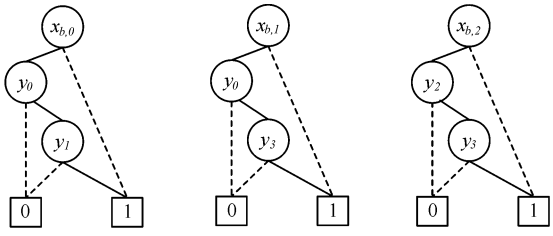
OBDD 为布尔函数的表示形式,故在 OBDD 表示 SAT 之前,需先将 SAT 中的约束子句转化为布尔函数。

例如,对于上述例子中的约束,公式(2)~公式(6)可以分别表示为布尔函数 $x'_{a,o}+y_2\cdot y_3$;$x'_{a,1}+y_1\cdot y_2\cdot y_3$;$x'_{b,o}+y_0\cdot y_1$;$x'_{b,1}+y_0\cdot y_3$;$x'_{b,2}+$

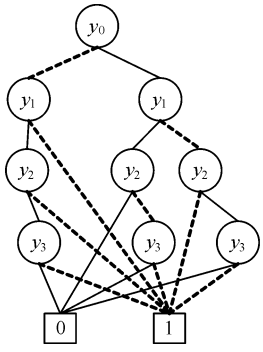
$y_2 \cdot y_3$ 其中“·”“′”和“+”分别表示布尔“与”“非”和“或”运算。由此可得所有约束子句对应的 OBDD 表示,如图 3 所示。



(a) 约束1 Constraint 1 (b)公式(2) Formula (2) (c)公式(3) Formula (3)



(d)公式(4) Formula (4) (e)公式(5) Formula (5) (f)公式(6) Formula (6)



(g)约束 3 Constraint 3

图 3 约束子句的符号 OBDD 表示

Fig. 3 Symbolic OBDD representation of the constraint clauses

2.3 基于 SAT 的 FPS 问题的符号 OBDD 算法

运用深度优先结合 OBDD 算法挖掘频繁序列。算法步骤如下:

Step 1:根据给定的序列以及最小支持度阈值,列举所有约束子句,并分别转化为布尔函数。将约束 1 表示为 OBDD。

Step 2:将 SAT 中约束 2 的变量根据该变量与其他变量之间约束关系的个数大小进行递增排序,然后将 SAT 中的约束按照此变量序进行归类,将每一类中的约束进行 OBDD 的“与”操作,最后对所有约束 2 以及约束 1 进行 OBDD 的“与”操作。

基于 SAT 的符号 OBDD 描述,求解 SAT 所有解的最直接方法就是利用 OBDD 的“与”操作,将 SAT 中的约束逐个的进行“与”运算,最后得到的 OBDD 即为 SAT 的所有解。但符号直接求解法在求解过程中,可能会使计算过程中生成的 OBDD 过于庞大,从而产生组合爆炸。而且基于 OBDD 的各种操作,其计算时间主要取决于参与操作的 OBDD 的大小。为改善算法的性能,先对 SAT 中的所有变量根据其在约束图中的度的大小进行递增排序,其中变量的度是指该变量与其他变量之间的约束关系的个数。假设按度的大小递增排序后的变量序为 $y_1 < y_2 < \cdots < y_n$,然后将 SAT 中的约束按照此变量序进行归类,即所有包含变量 y_1 的约束归为一类,然后在剩余的约束中将包含变量 y_2 的约束也归为一类,依此类推。先将每一类中的约束进行 OBDD 的“与”操作,然后对所有约束进行 OBDD 的“与”操作。

Step 3:将约束 3 表示为 OBDD,然后和约束 1,2“与”后的 OBDD 进行“与”操作,所得的 OBDD 即为满足所有约束的 SAT 的所有解,也即所给序列中相对于最小支持度阈值 λ 的所有频繁模式。图中的任意一条从根结点到终结点 1 的路径上取值为 1 的变量即为 SAT 的解,在路径中缺少的变量表示可以取 0 和 1 值。

3 实例分析

基于 DPLL 的基本思路,结合符号 OBDD 算法,对 FPS 问题的 SAT 模型进行求解,同时为验证该算法的正确性和可行性,运用实例进行分析,对于上述例子(序列 $aabb$ 的频繁模式挖掘问题),求解的主要步骤如下:

①列举所有约束子句,并转化为布尔函数,约束 1 的 OBDD 如图 3a 所示。

②对 SAT 中约束 2 的所有变量根据该变量与其他变量之间的约束关系的个数大小进行递增排序,排序后的变量序为 $x_{a,o} < x_{b,o} < x_{a,1} < x_{b,1} < x_{b,2} < y_0 < y_1 < y_2 < y_3$。

③将 SAT 中的约束按照此变量序进行归类:约束 (4)$x_{b,o} \rightarrow (y_0 \wedge y_1)$ 和约束(5)$x_{b,1} \rightarrow (y_0 \wedge y_3)$ 为一类;约束(3)$x_{a,1} \rightarrow (y_1 \wedge y_2 \wedge y_3)$ 为一类;约束(2)$x_{a,o} \rightarrow (y_2 \wedge y_3)$ 和约束(6)$x_{b,2} \rightarrow (y_2 \wedge y_3)$ 归为一类,并按类进行 OBDD 的“与”操作,然后再“与”上约束 1,由此可得约束 1“与”约束 2 的 OBDD 表示,如图 4 所示。

④将约束 3 表示为 OBDD,如图 3g 所示。对约束 1、约束 2、约束 3 进行 OBDD“与”操作,可得

SAT 所有解的 OBDD 表示(图 5),即所给序列中相对于最小支持度阈值 λ 的所有频繁模式。图中从根结点到终结点 1 的路径 $x'_{a,o}x_{b,o}x'_{a,1}x'_{b,1}x'_{b,2}y_0y_1$ 和 $x_{a,o}x'_{b,o}x'_{a,1}x'_{b,1}y_2y_3$ 分别表示频繁模式 $\{x_{b,o}\}$ 和 $\{\{x_{a,o}\},\{x_{a,o},x_{b,2}\}\}$,对应于模式 b 和 a,$a*b$。

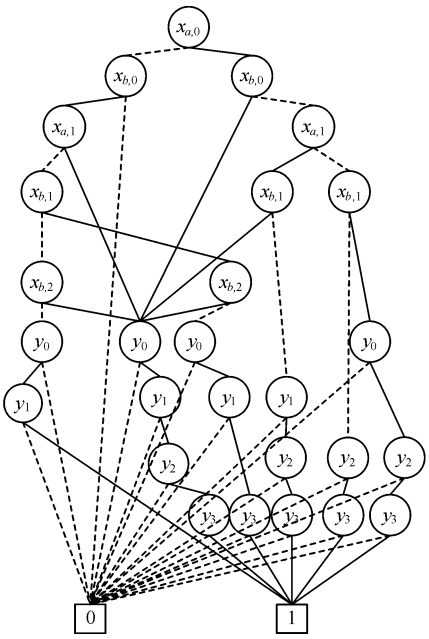


图 4 约束 1"与"约束 2 的 OBDD 表示

Fig. 4 OBDD representation of constraint 1 "and" constraint 2

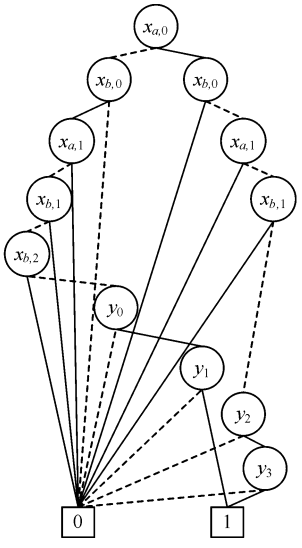


图 5 SAT 所有解的 OBDD 表示

Fig. 5 OBDD representation of all solutions of SAT

4 结论

本研究鉴于 OBDD 的高紧凑性和易操作性的特点,基于布尔可满足的思想与理论,以频繁序列挖掘问题为对象,探讨该问题的 SAT 模型以及相应的求解算法,减少了问题的空间需求,减缓了组合爆炸问题,得到了良好的研究结果。

数据挖掘是当前研究的主流方向。本研究只是在序列挖掘方面作了一些有益的探索工作,下一步研究方向和方法主要集中于两点:①将 SAT 模型以及 BDD 符号算法用于闭合、最大、最小频繁项集挖掘、关联规则挖掘等;②由于符号 OBDD 算法的求解时间主要依赖于 OBDD 图的大小,因此如何减小参与操作的 OBDD 图的大小,进一步提高算法的执行效率,是值得探究的方向之一。

参考文献:

[1] PEI J, HAN J, WANT W. Mining sequential patterns with constraints in large databases[C]. McLean Virginia: Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM), 2002: 18-25.

[2] MA Q, WANG J, SASHA D, et al. DNA sequence classification via an expectation maximizationalgorithm and neural networks: A case study[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2001, 31(4): 468-475.

[3] FERREIRA P G, AZEVEDO P J. Protein sequence classification through relevant sequences mining and bayes classifiers[C]//Portuguese Conference on Artificial Intelligence. [S. l.: s. n.], 2005: 236-247.

[4] DE RAEDT L, GUNS T, NIJSSEN S. Constraint programming for itemset mining[C]. New York, NY: KDD '08 Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, 2008: 204-212.

[5] GUNS T, NIJSSEN S, DE RAEDT L. Itemset mining: A constraint programming perspective[J]. Artif Intell: 2011, 175: 1951-1983.

[6] JABBOUR S, SAIS L, SALHI Y. Boolean satisfiability for sequence mining[C]. New York, NY: CIKM '13 Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, 2013: 649-658.

[7] PEI J, HAN J, MORTAZAVI-ASL B, et al. Mining sequential patterns by pattern-growth: The PrefixSpan approach[J]. IEEE Transactions on Knowledge and Data Engineering, 2004: 16(11): 1424-1440.

[8] LOEKITO E, BAILEY J, PEI J. A binary decision diagram based approach for mining frequent subsequences [J]. Knowledge & Information Systems, 2010, 24(2): 235-268.

(下转第 150 页 Continue on page 150)

CHANG L H. Application research on knowledge management in public health emergency decision support system taking Shanxi Province as an example[D]. Taiyuan:Taiyuan University of Technology,2013.

[5] 赵旭东,亚森·艾则孜. 基于互信息和余弦相似度的维吾尔文不良文档信息过滤方案[J]. 电子设计工程,2016,24(16):109-112.
ZHAO X D,YASEN A. A uyghur bad text information filtering scheme based on mutual information and cosine similarity[J]. Electronic Design Engineering,2016,24(16):109-112.

[6] 蒙杰,杨生举,施韶亭. 基于文本挖掘的科研项目管理辅助决策系统研究与实现[J]. 计算机应用与软件,2016,33(9):24-26,55.
MENG J,YANG S J,SHI S T. Study and implementation of text mining-based asistant decision support system for scientific research project management[J]. Computer Applications and Software,2016,33(9):24-26,55.

[7] 朱青,卫柯臻,丁兰琳,等. 基于文本挖掘和自动分类的法院裁判决策支持系统设计[J]. 中国管理科学,2018,26(1):170-178.
ZHU Q,WEI K Z,DING L L,et al. Design of court decision support system based on text mining and automatic classification[J]. Chinese Journal of Management Science,2018,26(1):170-178.

[8] 刘明昌. 基于内容的推荐技术研究[J]. 现代营销,2016,6:243.
LIU M C. Research on content based recommendation technology[J]. Marketing Management Review,2016,6:243.

[9] 杨武,唐瑞,卢玲. 基于内容的推荐与协同过滤融合的新闻推荐方法[J]. 计算机应用,2016,36(2):414-418.
YANG W,TANG R,LU L. News recommendation method by fusion of content-based recommendation and collaborative filtering[J]. Journal of Computer Applications,2016,36(2):414-418.

[10] 刘冰,李文书. 基于余弦相似度的指纹匹配算法的室内定位方法[J]. 科技通报,2017,33(3):198-202.
LIU B,LI W S. Indoor positioning method based on cosine similarity of fingerprint matching algorithm[J]. Bulletin of Science and Technology,2017,33(3):198-202.

[11] 李梦洁,邵曦. 基于文本属性的微博用户相似度研究[J]. 计算机技术与发展,2018,5. http://kns.cnki.net/kcms/detail/61.1450.TP.20180207.1917.082.html
LI M J,SHAO X. Research of micro-blog user similarity based on text similarity[J]. Computer Technology and Development,2018,05. http://kns.cnki.net/kcms/detail/61.1450.TP.20180207.1917.082.html

[12] 乔峰. 基于模板化网络爬虫技术的 Web 网页信息抽取[D]. 成都:电子科技大学,2012.
QIAO F. Web page information extraction based on formwork web crawler technology[D]. Chengdu:University of Electronic Science and Technology of China,2012.

[13] 武永亮,赵书良,李长镜,等. 基于 TF-IDF 和余弦相似度的文本分类方法[J]. 中文信息学报,2017,31(5):138-145.
WU Y L,ZHAO S L,LI C J,et al. Text classification method based on TF-IDF and cosine similarity[J]. Journal of Chinese Information Processing,2017,31(5):138-145.

[14] 董洋溢,李伟华,于会. 基于混合余弦相似度的中文文本层次关系挖掘[J]. 计算机应用研究,2017,34(5):1406-1409.
DONG Y Y,LI W H,YU H. Hierarchical relation mining of Chinese text based on mixed cosine similarity[J]. Application Research of Computers,2017,34(5):1406-1409.

(责任编辑:米慧芝 符支宏)

(上接第 142 页 Continue from page 142)

[9] CAMBAZARD H,HADZIC T,O'SULLIVAN B. Knowledge compilation for itemset mining[C]//Conference on ECAI 2010:European Conference on Artificial Intelligence. [S. l.]:IOS Press,2010:1109-1110.

[10] TSEITIN G. On the complexity of derivations in the propositional calculus[J]. Zap Nauchn Sem Lomi,1968:234-259.

[11] DAVIS M,LOGEMANN G,LOVELAND D W. A machine program for theorem-proving[J]. Communications of the ACM,1962,5(7):394-397.

[12] MARQUES-SILVA J P,SAKALLAH K A. GRASP—A new search algorithm for satisfiability[C]//IEEE/ACM International Conference on Computer-Aided Design,1996. Iccad-96. IEEE:[s. n.],1997:220-227.

[13] ZHANG L,MADIGAN C F,MOSKEWICZ M H,et al. Efficient conflict driven learning in a Boolean satisfiability solver[C]//IEEE/ACM International Conference on Computer-Aided Design. IEEE Press:[s. n.],2001:279-285.

[14] BRYANT R E. Graph-based algorithms for boolean function manipulation[J]. IEEE Trans Comput,1986,C-35(8):677-691.

[15] GU T L,XU Z B. Ordered binary decision diagram and itsapplication[M]. Beijing:Science Press,2009.

[16] BRYANT R E. Symbolic Boolean manipulation with ordered binary decision diagrams[J]. ACM Computing Surveys,1992,24(3):293-318.

(责任编辑:米慧芝)