

基于 BDD 的新因果图算法及在银行外部欺诈风险中的应用*

蒋 杰, 王洪春

(重庆师范大学 数学科学学院, 重庆 401331)

摘要:【目的】针对因果图传统推理方法中不交化割集的推理过程太过繁琐的问题,提出利用二元决策图(BDD)的推理方法。【方法】首先将因果图转化为标准化因果树,再转化为 BDD 结构,然后通过搜索路径直接得到简化的不交化割集。【结果】理论分析表明,该方法避开了求解最终割集的过程以及去除冗余的不交化割集的过程,从而降低了因果图推理中的计算复杂度。【结论】将新方法应用于银行外部欺诈风险实例中进行分析,通过与传统推理过程相比较,新方法更高效。
关键词:因果图;二元决策图;标准化因果树;不交化割集

中图分类号:TP38

文献标志码:A

文章编号:1672-6693(2018)06-0056-08

近些年来银行外部欺诈的事件频频发生,而银行系统关乎国民经济命脉,对其外部欺诈风险进行研究并做出相应的预防措施极其重要。本研究将利用基于 BDD 的因果图来对影响银行外部欺诈风险的因素进行研究分析。

动态不确定因果图是张勤教授^[1]于 1994 年提出的一种不确定性知识表达的推理模型。目前研究人员已经研究了各种模型的因果图,如单值因果图^[1]、多值因果图^[2-4]、连续因果图^[5]、模糊因果图^[6]等模型。并且近些年来把因果图应用于故障诊断已取得了一定的进展^[7-10],其中文献^[9-10]对因果图在化工故障诊断中的应用作了研究。文献^[11-12]分别把动态不确定因果图应用到了医药领域和核电领域。在进行故障诊断时,需要准确迅速地确定故障源,避免危险发生,但在复杂的系统模型中,传统的因果图分析容易导致组合爆炸的问题,不利于迅速定位故障源。

为了避免求解一阶割集和最终割集时出现组合爆炸的情况,本文提出了一种新的因果图分析方法,即利用二元决策图(Binary decision diagram, BDD)对因果图进行推理分析,该方法是在文献^[13]的基础上进行的完善处理。文献^[13]的主要思想是将因果图中的连接事件代替 BDD 结构中的 1 分支,进行遍历搜索得到不交化割集,这一方法在求解复杂因果图模型时,并不能得到正确结果。本文与文献^[13]的不同之处是保留了 BDD 结构中的 1 分支,且将连接事件作为 BDD 的中间节点放到了结构中。这一方法的主要思想是将因果图转化为以节点事件为根的标准化因果树,再将标准化因果树转化为 BDD 结构,从 BDD 结构的根节点搜索到终结点 1 的路径是一个不交化割集,遍历完所有能到达终结点 1 的路径得到节点事件所有的不交化割集,然后用逻辑运算符“+”将它们连接起来得到不交化割集表达式。采用 BDD 结构的推理降低了求解不交化割集的计算量,提高了推理速度。与文献^[13]处理方式比起来,本文的方法优点在于对较复杂的因果图模型进行推理时能得到更准确完整的结果。最后通过实例也证明了本文的方法比文献^[13]的方法得到的结果更准确。

1 因果图模型

因果图是一种用图形化的方式来表达事件之间因果关系的推理模型,在因果图中是允许有环路出现的。其中圆圈表示节点事件,是故障系统中可以观察到的故障现象;方框表示基本事件,是故障系统中的故障发生的原因;门结点表示逻辑门(通常表示的是逻辑或门或者逻辑与门,因为其它逻辑门都可以转化成这两种形式);有向

* 收稿日期:2018-01-15 修回日期:2018-10-10 网络出版时间:2018-10-25 10:42

资助项目:国家社会科学基金(No. 13BTJ008);重庆师范大学研究生科研创新项目(No. YKC17002)

第一作者简介:蒋杰,女,研究方向为因果图、不确定性推理,E-mail: 1016724527@qq.com;通信作者:王洪春,男,教授,博士,E-mail: wanghc@cqnu.edu.cn

网络出版地址: <http://kns.cnki.net/kcms/detail/50.1165.N.20181025.1042.022.html>

边表示连接事件,即原因事件导致结果事件的发生。

因果图中通常用大写字母表示事件,对应的小写字母表示事件发生的概率, X_i 表示节点事件, B_i 表示基本事件, G_i 表示逻辑门事件, V_i 表示上述 3 种事件的任意一种事件, P_{ij} 表示连接事件。图 1 为因果图示例。

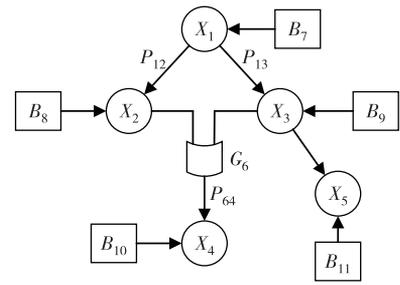


图 1 因果图示例

Fig. 1 An example of causality diagram

因果图推理步骤:1) 求节点事件的一阶割集表达式。

每个割集内事件之间的关系是逻辑与,即每个割集是由一系列事件(可以是基本事件、连接事件、节点事件、逻辑门事件)通过逻辑与的关系组合在一起构成的。割集之间的关系是逻辑或,仅与该节点事件相邻的事件构成的割集称为一阶割集。图 1 中各个节点事件的一阶割集表达式为:

$$X_1 = B_7, X_2 = B_8 \cup P_{12} X_1, X_3 = B_9 \cup P_{13} X_1, X_4 = B_{10} \cup P_{64} (X_2 \cup X_3), X_5 = B_{11} \cup P_{35} X_3。$$

2)求节点事件的最终割集表达式。

把节点事件的一阶割集表达式中的其它节点事件用基本事件代换掉,最后只剩下基本事件和连接事件的逻辑表达式称为该节点事件的最终割集。图 1 中各个节点事件的最终割集表达式为:

$$X_1 = B_7, X_2 = B_8 \cup P_{12} B_7, X_3 = B_9 \cup P_{13} B_7,$$

$$X_4 = B_{10} \cup P_{64} B_8 \cup P_{64} P_{13} B_7 \cup P_{64} P_{12} B_7 \cup P_{64} B_9, X_5 = B_{11} \cup P_{35} B_9 \cup P_{13} B_7。$$

3) 求节点事件的不变化割集表达式。

设 $X = \bigcup_{i=1}^m C_i$, 其中 $C_i = \bigcap_{j=1}^{n_i} V_{ij}$, C_i 是一个割集,则节点事件的不变化割集表达式可表示为:

$$X = C_1 + C_2 \overline{C_1} + C_3 \overline{C_1} \overline{C_2} + \dots + C_m \overline{C_1} \overline{C_2} \dots \overline{C_{m-1}},$$

其中,“+”为互斥或操作符。图 1 中各个节点事件的不变化割集表达式为:

$$X_1 = B_7, X_2 = B_8 + P_{12} B_7 \overline{B_8}, X_3 = B_9 + P_{13} B_7 \overline{B_9},$$

$$X_4 = B_{10} + P_{64} B_8 \overline{B_{10}} + P_{64} B_9 \overline{B_{10}} \overline{B_8} + P_{64} P_{12} \overline{B_{10}} B_9 \overline{B_8} + P_{64} P_{12} B_7 \overline{B_{10}} B_9 \overline{B_8} + P_{64} P_{13} B_7 \overline{B_{10}} B_9 \overline{B_8} P_{12},$$

$$X_5 = B_{11} + P_{35} B_9 \overline{B_{11}} + P_{35} P_{13} B_7 \overline{B_{11}} \overline{B_9}。$$

4) 求解在证据 E 下(证据 E 是事先给定的)计算感兴趣事件的后验概率 $P(V_i | E)$,首先通过逻辑吸收运算得到 E 和 $V_i E$ 的不变化割集表达式,再通过公式 $P(V_i | E) = \frac{P(V_i E)}{P(E)}$,直接带入数据得到后验概率。

定义 1 以因果图中节点事件为树根,以与节点事件相邻的父辈连接事件、基本事件、节点事件为树叶通过逻辑门构成的因果树称为标准化微因果树。

取因果树中的任意节点事件作根节点,若此节点事件有多个输入事件,则添加逻辑或门将这输入事件连接起来;若有向边用符号 P_{ij} 进行了标注,则添加逻辑与门将连接事件 P_{ij} 、基本事件、节点事件连接起来。将图 1 中的因果图转化为标准化微因果树,如图 2 所示。

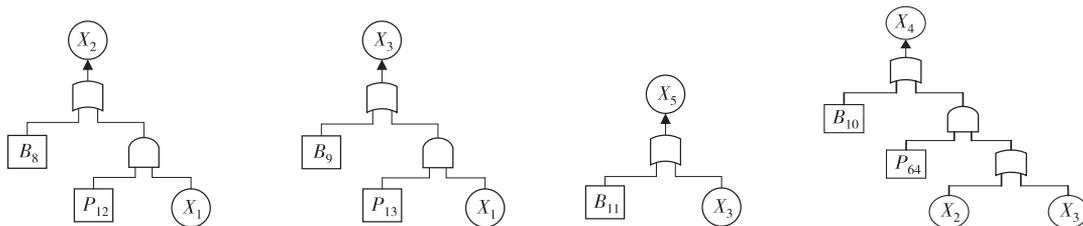


图 2 标准化微因果树示例

Fig. 2 Examples of standardized mini causality tree

定义 2 将标准化微因果树中的节点事件用基本事件、连接事件替换掉。此时,除了根节点外没有任何节点事件,即树叶只有基本事件、连接事件,称这样的因果树为标准化因果树。

以图 2 中节点事件 X_4 为例,转化成标准化因果树,如图 3 所示。

2 BDD 基本概念

BDD 是一种特殊的树型结构图^[14],采用二叉树的形式表示一个布尔逻辑函数,BDD 中的结点分为两类:一类结点是具有 0 或 1 两种布尔函数值的终结点;另一类结点是非终结点,没有确定的结点值,是内部结点。所有的结点都是通过具有 0 或 1 标识的边(称为 0 分支或 1 分支)连接在一起,从而组成一个完整的有向无环图。因果图转化为 BDD 结构时,通过逻辑与门输入的事件之间用 1 分支相连,通过逻辑或门输入的事件用 0 分支相连。

根据 Shannon 分解原理,把因果图中每个节点事件看作一个 ite(if-then-else)结构,ite(X, F_1, F_2)表示如果 X 成立,则考虑 F_1 ;否则考虑 F_2 。转化的递归算法思路是:从基本事件开始,用 ite 结构进行转化,用基本事件替换逻辑门,一层一层向下转化,直到所有的逻辑门都被基本事件替换,则节点事件的 BDD 形成。

令 $M = ite(X_i, A_1, A_2), N = ite(X_j, B_1, B_2)$,转化规则^[15]如下(其中 $\langle op \rangle$ 表示逻辑或或逻辑与操作):当 $X_i < X_j$ 时, $M \langle op \rangle N = ite(X_i, A_1 \langle op \rangle N, A_2 \langle op \rangle N)$;当 $X_i = X_j$ 时, $M \langle op \rangle N = ite(X_i, A_1 \langle op \rangle B_1, A_2 \langle op \rangle B_2)$ 。

3 基于 BDD 结构的因果图推理

用因果图对系统进行定量分析的关键就是求出不交化割集,而因果图中事件之间的“或”运算不是互斥的,所以需要转化为“互斥或”的关系。考虑到 BDD 中的分支之间是不相交的,而且能清晰地展示出影响节点事件状态的基本事件及传播路径,所以将因果图转化为 BDD 进行分析,运算过程将更高效。

用 BDD 进行不交化割集表达式推理的步骤:

- 1) 将各节点事件的因果树转化为标准化因果树;
- 2) 按深度优先最左遍历的顺序依次遍历,直到将标准化因果树遍历完为止,根据 Shannon 分解原理,形成节点事件对应的 BDD 结构;
- 3) 从 BDD 的根节点开始搜索到终结点 1 停止,保留整个路径中的基本事件和连接事件,事件之间是逻辑与的关系,若事件与 0 分支相连,则将事件改为对立事件,一条路径就是一个不交化割集,搜索完所有能到达终结点 1 的路径,得到所有的不交化割集;
- 4) 将步骤 3 中得到的不交化割集用互斥或操作符“+”组织起来,得到节点事件的不交化割集表达式。

定理 从 BDD 根节点到达终结点 1 的路径中,基本事件和连接事件按一定的方式排列成的因果链就是一个不交化割集。

证明 在标准化因果树中,节点事件的 BDD 结构形成的过程反应了基本事件、连接事件对节点事件影响的状态,以及影响节点事件的整个过程的传播路径,所以搜索路径时得到的基本事件和连接事件组成的因果链从本质上说就是一个不交化割集。在 BDD 结构中,从上一节点到达下一节点的路径中,若用 1 分支连接,则下一节点完全继承上一节点的信息,若用 0 分支连接,则下一节点继承上一节点的补信息,一直搜索到终结点 1 停止,得到一个不交化割集。由于 BDD 结构中的分支都是互斥的,所以得到的每条因果链之间是互斥或的关系,遍历完所有能到达终结点 1 的路径就可以直接得到节点事件的不交化割集表达式。证毕

将图 3 中的标准化因果树转化为 BDD 结构如图 4 所示。

从 B_{10} 开始搜索到达终结点 1 的路径,得到不交化割集为:

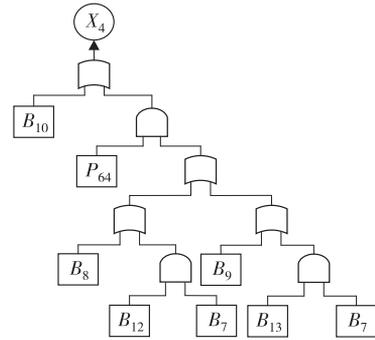


图 3 标准化因果树示例

Fig. 3 Examples of standardized causality tree

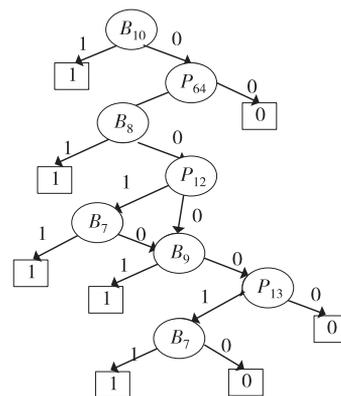


图 4 节点事件 X_4 对应的 BDD 结构

Fig. 4 A BDD structure corresponding to the node event X_4

$$B_{10}, P_{64} B_8 \overline{B_{10}}, P_{64} B_9 \overline{B_{10}} \overline{B_8}, P_{64} P_{12} B_7 \overline{B_{10}} \overline{B_9} \overline{B_8}, P_{64} P_{13} B_7 \overline{B_{10}} \overline{B_9} \overline{B_8} P_{12},$$

通过互斥或操作符“+”将它们连接起来得到 X_4 的不变化割集表达式:

$$X_4 = B_{10} + P_{64} B_8 \overline{B_{10}} + P_{64} B_9 \overline{B_{10}} \overline{B_8} + P_{64} P_{12} \overline{B_{10}} \overline{B_9} \overline{B_8} + P_{64} P_{12} B_7 \overline{B_{10}} \overline{B_9} \overline{B_8} + P_{64} P_{13} B_7 \overline{B_{10}} \overline{B_9} \overline{B_8} P_{12}.$$

和上文中第二节用因果图传统推理方法得到 X_4 的不变化割集表达式的推理过程相比,利用 BDD 来推理具有明显的优势。首先 BDD 结构更加清楚直观地展现出了事件之间的不变化关系,省略了求解一阶割集和最终割集的过程,降低了“组合爆炸”发生的概率;其次 BDD 结构分析不变化这一过程的计算量是呈线性的,耗时较短,降低了计算量,提高了推理速度;最后得到的不变化割集已经是最简化的,无需再进行吸收运算,减少了计算步骤,降低了计算复杂度。

用文献[13]的方法求得的 X_4 的不变化割集为 $B_{10}, P_{64} B_8 \overline{B_{10}}, P_{64} B_9 \overline{B_{10}} \overline{B_8}, P_{64} P_{12} B_7 \overline{B_{10}} \overline{B_9} \overline{B_8}$, 这个结果是不完整的。出现这一结果的原因是因为如果按照文献[13]的方法,会直接将 1 分支的 1 改为了连接事件,这样推理得到的结果是不会出现连接事件的补事件的,而这与实际不符,比如上述求 X_4 的不变化割集时,就遗漏掉了连接事件补事件这一情况,因此本文在文献[13]的基础上,进一步研究,优化推理过程,利用 BDD 进行因果图推理使得到的结果更准确完整。

此外利用 BDD 分析因果图还可以得出最小割集(如果去掉最终割集中的任一个基本事件后就不再是割集,则这个割集中的基本事件构成的集合称为最小割集)^[16],在得到的不变化割集基础上,除去不变化割集中的补事件以及连接事件,也就是在搜索路径的时候只保留非终结点 1 分支上的基本事件。可得节点事件 X_4 的最小割集为 B_{10}, B_9, B_8, B_7 。

将因果图转化为 BDD 结构进行遍历直接得到不变化割集,避免了在求解最终割集时组合爆炸的发生。同时通过 BDD 结构还可以得出最小割集,一次性得到了两方面的结果,使得推理过程更高效。

4 实例分析

以文献[17]中提供的数据(见表 1),利用文中给出的因果图推理的新方法对银行外部欺诈风险的实例进行分析,对应的因果图如图 5 所示。

表 1 各事件的名称及数据

Tab. 1 Names and data of events

事件名称	发生概率	事件名称	发生概率
B_1 违规操作	0.044 5	P_{41}	0.038 9
B_2 合谋欺骗	0.061 6	P_{12}	0.008 9
B_3 内部监控缺失	0.311 0	P_{22}	0.012 3
B_4 业务知识缺失	0.014 0	P_{32}	0.062 1
B_5 冒名顶替	0.297 0	P_{42}	0.002 7
B_6 无中生有	0.379 0	P_{53}	0.059 4
B_7 主动盗抢	0.218 0	P_{63}	0.075 8
B_8 转移资产	0.017 0	P_{73}	0.043 7
B_9 法律漏洞	0.130 0	P_{83}	0.003 4
B_{10} 信用缺失	0.065 0	P_{94}	0.025 9
P_{21}	0.086 0	P_{104}	0.013 0
P_{31}	0.182 3		

图 5 中 X_1 表示银行外部欺诈风险; X_2 表示银行内部监控; X_3 表示客服; X_4 表示金融环境。

对上述问题的不变化割集表达式的求解,下文将分别用传统推理方法、文献[13]的方法、本文提出的方法进行对比说明。

根据因果图传统推理方法对 X_1 进行推理,过程如下。

首先求一阶割集为:

$$X_1 = P_{21} X_2 \cup P_{31} X_3 \cup P_{41} X_4,$$

$$X_2 = P_{12} B_1 \cup P_{22} B_2 \cup P_{32} B_3 \cup P_{42} B_4,$$

$$X_3 = P_{53} B_5 \cup P_{63} B_6 \cup P_{73} B_7 \cup P_{83} B_8, X_4 = P_{94} B_9 \cup P_{104} B_{10};$$

其次求 X_1 最终割集为:

$$X_1 = P_{21} (P_{12} B_1 \cup P_{22} B_2 \cup P_{32} B_3 \cup P_{42} B_4) \cup$$

$$P_{31} (P_{53} B_5 \cup P_{63} B_6 \cup P_{73} B_7 \cup P_{83} B_8) \cup P_{41} (P_{94} B_9 \cup P_{104} B_{10});$$

然后求出 X_1 的不变化割集,再将最终割集中的“或”运算符转化为“互斥或”运算符之后,得到的不变化割集表达式还不是

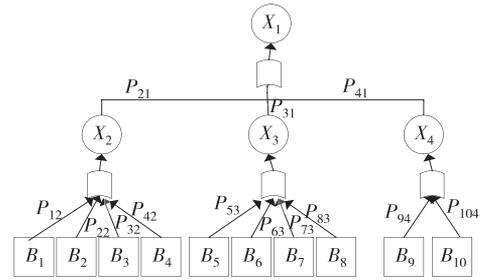


图 5 银行外部欺诈因果图

Fig. 5 A causality diagram of bank external fraud

最简的,需要反复进行逻辑吸收运算,因为篇幅所限,下面直接给出已经化简了的不变化割集表达式:

$$\begin{aligned}
X_1 = & P_{21} P_{12} B_1 + P_{21} P_{22} B_2 \overline{P_{12}} + P_{21} P_{22} B_2 P_{12} \overline{B_1} + P_{32} P_{21} B_3 (\overline{P_{12}} + P_{12} \overline{B_1}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{12}} + P_{12} \overline{B_1}) + \\
& P_{31} P_{53} B_5 [\overline{P_{21}} + P_{21} (\overline{P_{42}} + P_{42} \overline{B_4}) + P_{21} P_{42} B_4 (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{12}} + P_{12} \overline{B_1})] + \\
& P_{31} P_{63} B_6 (\overline{P_{53}} + P_{53} \overline{B_5}) [\overline{P_{21}} + P_{21} (\overline{P_{42}} + P_{42} \overline{B_4}) (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{12}} + P_{12} \overline{B_1})] + \\
& P_{31} P_{73} B_7 (\overline{P_{63}} + P_{63} \overline{B_6}) (\overline{P_{53}} + P_{53} \overline{B_5}) [\overline{P_{21}} + P_{21} (\overline{P_{42}} + P_{42} \overline{B_4}) (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{12}} + P_{12} \overline{B_1})] + \\
& P_{31} P_{83} B_8 (\overline{P_{73}} + P_{73} \overline{B_7}) (\overline{P_{63}} + P_{63} \overline{B_6}) (\overline{P_{53}} + P_{53} \overline{B_5}) [\overline{P_{21}} + P_{21} (\overline{P_{42}} + P_{42} \overline{B_4}) (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{12}} + P_{12} \overline{B_1})] + \\
& P_{41} P_{94} B_9 [\overline{P_{31}} + P_{31} (\overline{P_{83}} + P_{83} \overline{B_8}) (\overline{P_{73}} + P_{73} \overline{B_7}) (\overline{P_{63}} + P_{63} \overline{B_6}) (\overline{P_{53}} + P_{53} \overline{B_5})] [\overline{P_{21}} + P_{21} (\overline{P_{42}} + P_{42} \overline{B_4}) [\overline{P_{21}} + \\
& P_{21} (\overline{P_{42}} + P_{42} \overline{B_4}) (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{12}} + P_{12} \overline{B_1})] + P_{41} P_{104} B_{10} (\overline{P_{94}} + P_{94} \overline{B_9}) [\overline{P_{31}} + \\
& P_{31} (\overline{P_{83}} + P_{83} \overline{B_8}) (\overline{P_{73}} + P_{73} \overline{B_7}) (\overline{P_{63}} + P_{63} \overline{B_6}) (\overline{P_{53}} + P_{53} \overline{B_5})] [\overline{P_{21}} + \\
& P_{21} (\overline{P_{42}} + P_{42} \overline{B_4}) (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{12}} + P_{12} \overline{B_1})].
\end{aligned}$$

由于图 5 已经是因果树的形式,所以直接按照文献[13]的方法得出对应的 BDD 结构,如图 6 所示:

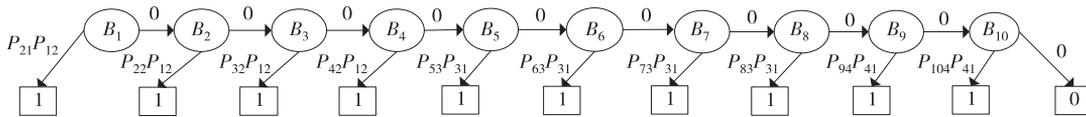


图 6 图 5 对应的 BDD 结构

Fig. 6 The BDD structure corresponding to Fig. 5

由图 6 进行遍历,得到 X_1 的不变化割集表达式如下所示:

$$\begin{aligned}
X_1 = & P_{21} P_{12} B_1 + P_{21} P_{22} B_2 \overline{B_1} + P_{21} P_{32} B_3 \overline{B_2} \overline{B_1} + P_{21} P_{42} B_4 \overline{B_3} \overline{B_2} \overline{B_1} + P_{31} P_{53} B_5 \overline{B_4} \overline{B_3} \overline{B_2} \overline{B_1} + \\
& P_{31} P_{63} B_6 \overline{B_5} \overline{B_4} \overline{B_3} \overline{B_2} \overline{B_1} + P_{31} P_{73} B_7 \overline{B_6} \overline{B_5} \overline{B_4} \overline{B_3} \overline{B_2} \overline{B_1} + P_{31} P_{83} B_8 \overline{B_7} \overline{B_6} \overline{B_5} \overline{B_4} \overline{B_3} \overline{B_2} \overline{B_1} + \\
& P_{41} P_{94} B_9 \overline{B_8} \overline{B_7} \overline{B_6} \overline{B_5} \overline{B_4} \overline{B_3} \overline{B_2} \overline{B_1} + P_{41} P_{104} B_{10} \overline{B_9} \overline{B_8} \overline{B_7} \overline{B_6} \overline{B_5} \overline{B_4} \overline{B_3} \overline{B_2} \overline{B_1}.
\end{aligned}$$

通过传统推理方法得到的结果可以看出,文献[13]的方法得到的结果是错误的,文献[13]的方法错误的原因为默认了连接事件与各个事件之间是互斥的关系,这与实际情况是相矛盾的,所以导致了错误的结果。

接下来利用本文介绍的方法求解不变化割集表达式。因为图 5 已经是因果树的形式所以可以直接将图 5 转化为标准化因果树,如图 7 所示。

将得到的标准化因果树转化为 BDD 结构如图 8 所示。

由于此问题的 BDD 结构较复杂,所以在遍历的时候,没有一条一条的遍历,而是一层一层遍历,但是实质没有变化。以遍历到 B_3 的终结点 1 为例进行说明,从 P_{21} 开始只有一条路径到达 B_3 ,遍历到 P_{12} 有两条路径到达 B_3 ,遍历到 P_{22} 有两条路径到达 B_3 ,遍历到 P_{32} 有一条路径到达 B_3 ,所以能到达 B_3 的路径条数为 $1 \times 2 \times 2 \times 1 = 4$,用事件表示为: $P_{21} (\overline{P_{12}} + P_{12} \overline{B_1}) (\overline{P_{22}} + P_{22} \overline{B_2}) P_{32} B_3$,这就是遍历到 B_3 时得到的不变化割集,由于篇幅所限后面得到的不变化割集都不将括号去掉。

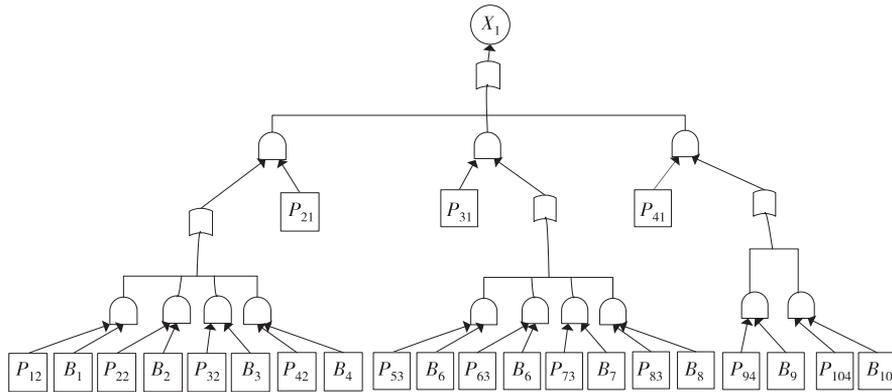


图 7 图 5 所对应的标准化因果树

Fig. 7 The standardized causality tree corresponding to Fig. 5

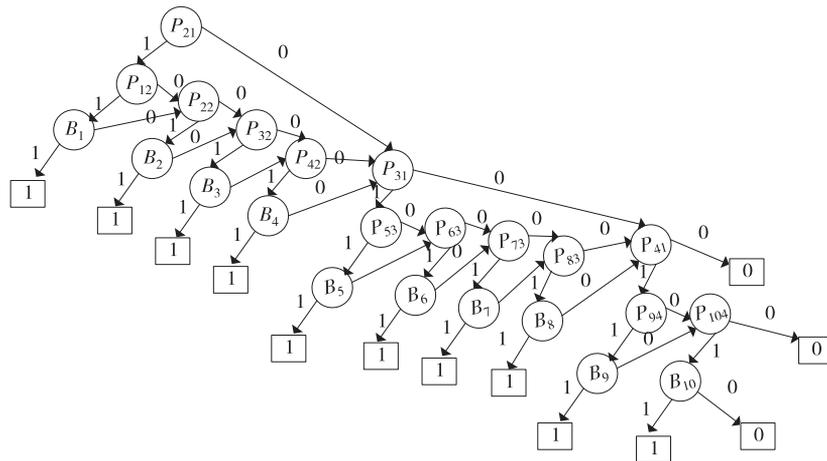


图 8 图 7 所对应的 BDD 结构

Fig. 8 The BDD structure corresponding to Fig. 7

根据图 8 的 BDD 结构只需遍历 10 次就可以得出 X_1 的不交化割集表达式为:

$$\begin{aligned}
 X_1 = & P_{21} P_{12} B_1 + P_{21} P_{22} B_2 (\overline{P_{12}} + P_{12} \overline{B_1}) + P_{21} (\overline{P_{12}} + P_{12} \overline{B_1}) (\overline{P_{22}} + P_{22} \overline{B_2}) P_{32} B_3 + \\
 & P_{21} (\overline{P_{12}} + P_{12} \overline{B_1}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{32}} + P_{32} \overline{B_3}) P_{42} B_4 + \\
 & [\overline{P_{21}} + P_{21} (\overline{P_{42}} + P_{42} \overline{B_4}) (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{21}} + P_{21} \overline{B_1})] P_{31} P_{53} B_5 + \\
 & [\overline{P_{21}} + P_{21} (\overline{P_{42}} + P_{42} \overline{B_4}) (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{21}} + P_{21} \overline{B_1})] (\overline{P_{53}} + P_{53} \overline{B_5}) P_{31} P_{63} B_6 + \\
 & [\overline{P_{21}} + P_{21} (\overline{P_{42}} + P_{42} \overline{B_4}) (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{21}} + P_{21} \overline{B_1})] (\overline{P_{63}} + P_{63} \overline{B_6}) (\overline{P_{53}} + P_{53} \overline{B_5}) P_{31} P_{73} B_7 + \\
 & [\overline{P_{21}} + P_{21} (\overline{P_{42}} + P_{42} \overline{B_4}) (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{21}} + P_{21} \overline{B_1})] (\overline{P_{73}} + \\
 & P_{73} \overline{B_7}) (\overline{P_{63}} + P_{63} \overline{B_6}) (\overline{P_{53}} + P_{53} \overline{B_5}) P_{31} P_{83} B_8 + \\
 & [\overline{P_{21}} + P_{21} (\overline{P_{42}} + P_{42} \overline{B_4}) (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{21}} + P_{21} \overline{B_1})] [\overline{P_{31}} + P_{31} (\overline{P_{83}} + \\
 & P_{83} \overline{B_8}) (\overline{P_{73}} + P_{73} \overline{B_7}) (\overline{P_{63}} + P_{63} \overline{B_6}) (\overline{P_{53}} + P_{53} \overline{B_5})] P_{41} P_{94} B_9 + [\overline{P_{21}} + P_{21} (\overline{P_{42}} + \\
 & P_{42} \overline{B_4}) (\overline{P_{32}} + P_{32} \overline{B_3}) (\overline{P_{22}} + P_{22} \overline{B_2}) (\overline{P_{21}} + P_{21} \overline{B_1})] [\overline{P_{31}} + P_{31} (\overline{P_{83}} + P_{83} \overline{B_8}) (\overline{P_{73}} + P_{73} \overline{B_7}) [\overline{P_{31}} + \\
 & P_{31} (\overline{P_{83}} + P_{83} \overline{B_8}) (\overline{P_{73}} + P_{73} \overline{B_7}) (\overline{P_{63}} + P_{63} \overline{B_6}) (\overline{P_{53}} + P_{53} \overline{B_5})] (\overline{P_{94}} + P_{94} \overline{B_9}) P_{41} P_{104} B_{10}。
 \end{aligned}$$

通过 BDD 结构得到的不交化割集表达式已经是最简形式,不必再进行化简。与传统方法相比,推理过程更直观、更高效,不用进行逻辑吸收运算、不交化处理等一系列逻辑运算,所以推理过程也没有传统推理方法繁琐。

下面带入数据求出各个基本事件的后验概率:

$$\begin{aligned}
 P(B_1 | X_1) = & 0.046\ 842, P(B_2 | X_1) = 0.066\ 263, P(B_3 | X_1) = 0.398\ 198, P(B_4 | X_1) = 0.014\ 138, \\
 P(B_5 | X_1) = & 0.512\ 546, P(B_6 | X_1) = 0.622\ 155, P(B_7 | X_1) = 0.228\ 23, P(B_8 | X_1) = 0.008\ 7,
 \end{aligned}$$

$$P(B_9 | X_1) = 0.069\ 758, P(B_{10} | X_1) = 0.059\ 271。$$

由上述得到的后验概率可以看出,伪造票据、存折等无中生有行为,是造成银行外部欺诈风险的首要因素;其次是冒名顶替行为。为了降低风险发生的概率,银行系统应该提高防伪技术,以及在客户办理业务时,加强核实客户的资料信息的管理。

5 结束语

将求解不交化割集表达式推理计算过程,转化到 BDD 结构中,通过图形的形式将传统的推理过程呈现出来,使得整个推理过程更直观清楚。可以得出以下结论:

1) 与传统的因果图推理方法相比,避开了求解一节割集、最终割集的步骤,直接得到不交化割集,而且得到的不交化割集已经经过化简处理,不需要再进行逻辑吸收运算,减少了计算复杂度,提高了推理的速度。整个推理过程优于因果图传统的推理过程。

2) 文中提出的方法是在文献[13]的基础上进行的改进处理,通过第四节以及上述实例对比发现,弥补了文献[13]中方法的缺陷,使得到的结果更准确。

3) 通过实例可以看出在复杂的因果图模型中,利用文中提出的方法比传统推理方法推理速度更快,思路更清晰。

参考文献:

- [1] ZHANG Q. Probabilistic based on dynamic causality trees diagrams[J]. Reliability Engineering and System Safety, 1994, 46: 209-220.
- [2] 樊兴华,张勤,孙茂松,等. 多值因果图的推理算法研究[J]. 计算机学报, 2003, 26(3): 310-322.
FAN X H, ZHANG Q, SUN M S, et al. Reasoning Algorithm in multi-value causality diagram[J]. Chinese Journal of computers, 2003, 26(3): 310-322.
- [3] 董春玲,张勤. 用于不确定性故障诊断的权重逻辑推理算法研究[J]. 自动化学报, 2014, 40(12): 2766-2781.
DONG C L, ZHANG Q. Research on weighted logic inference algorithm for uncertain fault diagnosis[J]. Automation Journal, 2014, 40(12): 2766-2781.
- [4] 刘绍红,王洪春. 一种基于改进的多值因果图的近似推理[J]. 重庆理工大学学报(自然科学版), 2016, 30(4): 127-131.
LIU S H, WANG H C. Approximate reasoning based on improved multi-valued causal graph[J]. Journal of Chongqing University of Technology(Natural Science), 2016, 30(4): 127-131.
- [5] 王洪春. 连续因果图的推理及参数学习[J]. 微电子学与计算机, 2007, 24(1): 18-24.
WANG H C. Reasoning and learning of the parameter is continuous causality diagram[J]. Microelectronics & Computer, 2007, 24(1): 18-24.
- [6] 梁新元. 单值模糊因果图混合推理研究[J]. 微电子学与计算机, 2008, 25(2): 14-16.
LIANG X Y. Mixed reasoning of single-value fuzzy causality diagram [J]. Microelectronics & Computer, 2008, 25(2): 14-16.
- [7] 梁帆,王洪春. 基于因果图基本事件矩阵的研究[J]. 重庆师范大学学报(自然科学版), 2015, 32(3): 119-123.
LIANG F, WANG H C. The method by using basic event matrix of causality diagram[J]. Journal of Chongqing Normal University(Natural Science), 2015, 32(3): 119-123.
- [8] 张勤. DUCG: 一种新的动态不确定因果知识的表达和推理方法(I): 离散, 静态, 证据确定和有向无环图情况[J]. 计算机学报, 2010, 33(4): 625-651.
ZHANG Q. DUCG: a new methodology to deal with dynamical uncertain causalities (I): the static discrete DAG case[J]. Chinese Journal of Computers, 2010, 33(4): 625-651.
- [9] 杨佳婧,张勤,朱群雄. 动态不确定因果图在化工过程故障诊断中的应用[J]. 智能系统学报, 2014, 9(2): 154-160.
YANG J J, ZHANG Q, ZHU Q X. Application of dynamic uncertain causality graph to fault diagnosis in chemical processes[J]. CAAI Transactions on Intelligent Systems, 2014, 9(2): 154-160.
- [10] 曲彦光,张勤,朱群雄. 动态不确定因果图在化工过程故障诊断中的应用[J]. 智能系统学报, 2015, 10(3): 354-360.
QU Y G, ZHANG Q, ZHU Q X. Application of dynamic uncertain causality graph to fault diagnosis in chemical processes[J]. CAAI Transactions on Intelligent Systems, 2015, 10(3): 354-360.
- [11] DONG C L, WANG Y J, ZHANG Q, et al. The methodology of dynamic uncertain causality graph for intelligent diagnosis of vertigo[J]. Computer Methods and Programs

- in Biomedicine, 2014, 113(1):162-174.
- [12] ZHANG Q, DONG C L, CUI Y, et al. Dynamic uncertain causality graph for knowledge representation and probabilistic reasoning; statistics base, matrix, and application[J]. IEEE Transactions on Neural Networks and Learning Systems, 2014, 25(4):645-663.
- [13] 梁帆, 王洪春. 利用改进的二元决策图的因果图推理[J]. 计算机工程与应用, 2015, 51(18):229-232.
- LIANG F, WANG H C. Causality diagram reasoning by using binary decision diagram improvement[J]. Computer Engineering and Applications, 2015, 51(18):229-232.
- [14] 郎荣玲. 故障树转化为二元决策树的算法研究[J]. 计算机工程与应用, 2008, 44(27):69-71.
- LANG R L. Research on fault tree conversion to binary decision diagram[J]. Computer Engineering and Applications, 2008, 44(27):69-71.
- [15] 罗航, 王厚军, 黄建国, 等. 用递归 BDD 技术分析故障树[J]. 电子科技大学学报, 2011, 40(5):726-731.
- LUO H, WANG H J, HUANG J G, et al. Analyzing fault by using method of recursive BDD technique[J]. Journal of University of Electronic Science and Technology of China, 2011, 40(5):726-731.
- [16] 梁新元. 基于因果图最小割集的故障分析方法研究[J]. 微电子学与计算机, 2005, 22(1):92-97.
- LIANG X Y. Research on diagnosis approach based on MCS of causality diagram[J]. Microelectronics & Computer, 2005, 22(1):92-97.
- [17] 梁帆. 因果图推理的改进及应用[D]. 重庆:重庆师范大学, 2016.
- LIANG F. The application and reasoning improvement of causality diagram[D]. Chongqing:Chongqing Normal University, 2016.

A New Causality Diagram Inference Algorithms Based on BDD and Its Application in Bank External Fraud Risk

JIANG Jie, WANG Hongchun

(College of Mathematics, Chongqing Normal University, Chongqing 401331, China)

Abstract: [Purposes] The inference process to use the traditional method of causality diagram to calculate disjoint cut sets is too cumbersome, so the reasoning method using binary decision diagram (BDD) is proposed. [Methods] Firstly, the causality diagram is transformed into a standardized causality tree, and then into BDD structure. After that, those disjoint cut sets which have been simplified are obtained by searching path. [Findings] The theoretical analysis shows that the method avoids the process of solving the final cut set as well as removing redundant disjoint cut sets, thereby reducing computational complexity in causality diagram inference. [Conclusions] Finally, applied in instance of Bank external fraud risk, by compared with the traditional reasoning, this method is more efficient.

Keywords: causality diagram; binary decision diagram; standardized causality diagram; disjoint cut sets

(责任编辑 许 甲)