

# 高效计算因果网中的干预



李超<sup>1</sup> 覃飙<sup>2</sup>

1 中国政法大学商学院 北京 100088

2 中国人民大学信息学院 北京 100872

**摘要** 在因果网中,对和积问题因果效果的计算是其首要问题,从有向无环图的角度,研究者们发现每一个因果网都有一个与之对应的贝叶斯网络,干预是因果网的一个基本操作。类似于贝叶斯网络中的剪枝策略,在剪枝掉所有无效结点后,文中设计了一种优化的算法 OFDo 来计算对因果网中每个结点的完全原子干预。文中接着研究多干预操作,发现多干预操作具有可交换性,并基于多干预操作的可交换性证明了多干预操作的优化计算策略。最后,通过实验证实 OFDo 计算对因果网中所有结点完全原子干预的效率比目前的算法都好。

**关键词:** 因果网;干预;无效结点;完全原子干预;多干预

**中图法分类号** TP311

## Efficient Computation of Intervention in Causal Bayesian Networks

LI Chao<sup>1</sup> and QIN Biao<sup>2</sup>

1 Business School, China University of Political Science and Law, Beijing 100088, China

2 Information School, Renmin University of China, Beijing 100872, China

**Abstract** In causal Bayesian networks (CBNs), it is a fundamental problem to compute the causal effect of sum product. From the perspective of a directed acyclic graph, we show every CBN has a corresponding Bayesian network. Intervention is a fundamental operation in CBNs. Similar to Bayesian networks, CBNs also have the pruning strategy. After pruning the barren nodes, this paper devises an optimized jointree algorithm to compute the full atomic intervention on each node in a CBN. Then, this paper explores the multiple interventions on multiple nodes, and finds that multiple interventions have the commutative property. On the basis of the commutative property in multiple interventions, this paper proves the strategies, which can be used to optimize the computation of the causal effect of multiple interventions. Finally, we report experimental results to demonstrate the efficiency of our algorithm to compute the causal effects in CBNs.

**Keywords** Causal Bayesian networks, Intervention, Barren nodes, Full atomic intervention, Multiple interventions

## 1 引言

贝叶斯网络是一类概率图模型,它能够很自然和高效地描述不确定性知识。它由两部分组成:用有向无环图来表示结点之间的依赖以及用条件概率表表示结点之间的依赖程度。因果网(CBN)<sup>[1]</sup>采用贝叶斯网络来描述因果信息和传统的条件依赖,它们通过概率演算来计算因果效果,因此贝叶斯网络是因果网的基础。

有两种类型的干预在因果网领域进行了广泛的研究,它们分别是对一个结点的原子干预和对多个结点的多干预<sup>[1]</sup>。因为每个结点  $V_i$  有  $|V_i|$  个值,所以必须对该结点执行  $|V_i|$  次

干预来得到  $|V_i|$  个干预的结果。从这些干预的结果中,我们能够计算出对该结点的最大干预值。在这种应用的需求下,一种名为偏原子干预的新干预<sup>[2]</sup>被提出,它表示对一个结点上的一个原子干预和多原子干预。接着一个偏干预表(PIT)被提出来编码偏原子干预,因此,本文在因果网中计算干预效果时可以把条件概率表(CPT)替换为 PIT。

一个叶结点如果既不属于查询变量也不属于证据,那么它被称为无效结点,它对贝叶斯网络的边际概率是没有贡献的<sup>[3]</sup>。采用和贝叶斯网络中相同的方法,在因果网中也定义了无效结点<sup>[1]</sup>。假如我们对一个无效结点  $V_i$  进行干预,那么证据和查询变量的联合概率是独立于干预结点  $V_i$  的,也就是

到稿日期:2021-03-02 返修日期:2021-06-14 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:中国政法大学科研创新项目(19ZFG79002);国家自然科学基金项目(61772534);教育部哲学社会科学重大课题(19JHQ007);中国政法大学新兴学科培育与建设计划;中央高校基本科研业务费专项资金

This work was supported by the Scientific Research and Innovation Project of China University of Political Science and Law(19ZFG79002), National Science Foundation of China(61772534), Research Foundation from Ministry of Education of China(19JHQ007), New Discipline Construction Project of China University of Political Science and Law and Fundamental Research Funds for the Central Universities.

通信作者:李超(chaoli@cupl.edu.cn)

说干预是没有意义的。例如,图 1 中,有  $P(c|do(B))=P(c)$ 。在删除这些无效结点后,本文提出了一种优化的联合树算法来高效地计算所有结点的完全原子干预。残缺图和增广图不仅可以用来描述对单个结点的干预,还可以用来描述对多个结点的并行或者顺序干预<sup>[1]</sup>。接着,本文发现多干预具有可交换的特性,基于多干预的可交换性,本文证明了计算多干预的优化计算策略。这些研究有很多现实的应用,例如,去年新冠肺炎出现后,政府应该如何进行政策干预,才能实现既能防止新冠肺炎疫情的传播又能促进国民经济稳步增长的目标。

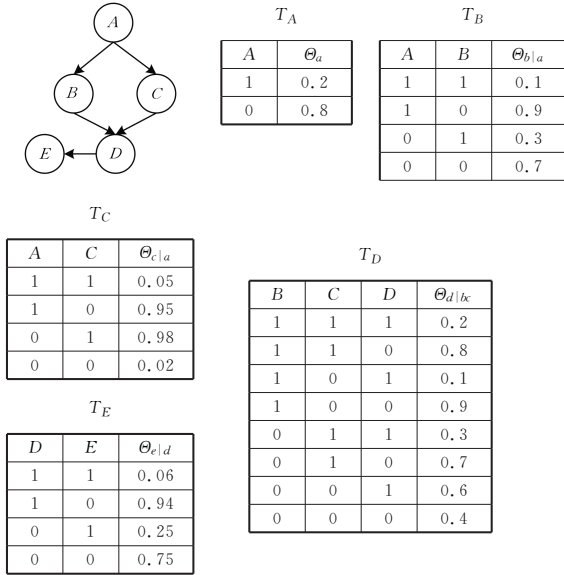


图 1 具有 5 个结点的贝叶斯网络

Fig. 1 Bayesian network with five nodes

本文的主要贡献如下:1)本文提出了因果网中完全原子干预的优化算法;2)本文发现了多干预的可交换性,并给出了计算多干预的优化策略。本文第 2 节概述了因果网的一些基础知识;第 3 节提出了一种联合树算法来高效地计算完全原子干预;第 4 节给出了多干预的可交换性和一种优化策略;第 5 节描述了实验结果;最后总结全文。

## 2 因果网的基本知识

### 2.1 贝叶斯网络的经典问题

在贝叶斯网络中,一条边  $E$  表示两个结点之间的条件依赖关系,每个结点有一个条件概率表(CPT)。在每个条件概率表中,我们观察到对于每一组父结点  $U_i$ ,都有  $\sum_i P(v_i | u_i) = 1$ 。我们把每一个集合  $V_i U_i$  看作一个网络家族,每个贝叶斯网络都隐含地表示一组变量  $V$  的联合概率分布。在给定父结点的情况下,每个结点条件都独立于它的祖先结点。在贝叶斯网络中推理过程包括两部分:网络和查询。从网络结构的角度看,一个贝叶斯网络对于联合概率分布具有条件独立的特性。从查询结构的角度看,我们可以从贝叶斯网络中逐步删除无效结点,在剪枝掉所有无效结点后,可以从这个剪枝过的贝叶斯网络中计算边际概率<sup>[3]</sup>。因为剪枝操作是递归进行的,通常可以剪枝掉很多无效结点,所以这个剪枝操作是非常实用的。

贝叶斯网络的联合树是一个对偶  $(\Gamma, C)$ ,这里表示联合

树,  $C = \{C_1, \dots, C_n\}$  是  $\Gamma$  中的一组结点,每个结点  $C_i$  也被称作一个聚簇。一个联合树必须满足以下 3 个特性:1)  $\Gamma$  中的每个结点  $C_i$  都是贝叶斯网络中的一组结点;2) 贝叶斯网络中的每个家族一定出现在某个结点  $C_j$  中;3) 假如贝叶斯网络的某个结点同时出现在两个聚簇结点  $C_i$  和  $C_j$  中,那么它也一定出现在从  $C_i$  到  $C_j$  路径中的每个结点  $C_k$  中。在  $\Gamma$  中从  $C_i$  到  $C_j$  的边称为分隔符  $S_{ij}$ ,可以定义  $S_{ij} = C_i \cap C_j$ 。联合树的大小可以定义为最大的聚簇减 1。给定某个证据  $e$ ,联合树算法在  $\Gamma$  中的结点间传递消息。对于每个结点  $C_i$  完成上行和下行的消息传递后,我们可以计算它的边际概率  $P(C_i, e)$ 。例如,计算图 1 中贝叶斯网络的边际概率  $P(a=0, e=0)$  的联合树,如图 2 所示,这里聚簇  $C_1$  包括结点 A, B 和 C;聚簇  $C_2$  包括结点 D;聚簇  $C_3$  包括结点 E。我们选择  $C_3$  作为根结点。消息的上行传递就是把各个结点的消息传递到  $C_3$ ,消息的下行传递就是把  $C_3$  的消息向其他结点进行传递。

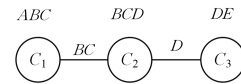


图 2 图 1 中贝叶斯网络的联合树

Fig. 2 Jointtree for Bayesian network in Fig. 1

Park-Darwiche 结构<sup>[4]</sup>结合 Shenoy-Shafer 的结构<sup>[5]</sup>和 Hugin 结构<sup>[6]</sup>,在该结构中传到同一个聚簇中的消息只需要相乘一次,并且可以采用一个除操作来计算偏导。Park-Darwiche 结构采用零意识表,首先回顾零意识数,它是一个对偶  $(z, b)$ ,这里  $z$  是一个标量,  $b$  是一个位数据,它有两个值: T (真)和 F (假)。考虑一组数  $x_1, \dots, x_n$ ,并且假设对于每个  $i = 1, \dots, n$ ,我们将计算  $m_i = \prod_{j \neq i} x_j$ 。假设  $z = \prod_{j=1}^n x_j$ , Park-Darwiche 结构定义了 4 种操作。首先它定义了如何从对偶  $(z, b)$  中计算  $m_i$ :如果  $b = F$ ,则  $m_i = z/x_i$ ;如果  $b = T$  并且  $x_i \neq 0$ ,则  $m_i = 0$ ;如果  $b = T$  并且  $x_i = 0$ ,则  $m_i = z$ 。其次,它定义了标量  $c$  和零意识数  $(z, b)$ :如果  $b = F$  并且  $c = 0$ ,则有  $(z, b) * c = (z, T)$ ;在其他情况下,  $(z, b) * c = (c * z, b)$ 。然后,它给出了两个零意识数的加法规则:如果  $b_1 = F$  并且  $b_2 = T$ ,则有  $(z_1, b_1) + (z_2, b_2) = (z_1, b_1)$ ;如果  $b_1 = T$  并且  $b_2 = F$ ,则有  $(z_1, b_1) + (z_2, b_2) = (z_2, b_2)$ ;在其他情况下,  $(z_1, b_1) + (z_2, b_2) = (z_1 + z_2, b_1)$ 。最后,一个操作定义为:如果  $b = F$ ,则有  $real(z, b) = z$ ;其他情况为  $real(z, b) = 0$ 。

类似地, Park-Darwiche 结构定义了零意识表,假如  $\Psi$  是零意识表,则  $\Phi$  表示标准表。Park-Darwiche 结构定义了边际值  $\sum_X \Psi$ 、积  $\Psi\Phi$ 、除  $\Psi/\Phi$  和  $real(\Psi)$ 。采用 Park-Darwiche 结构,我们可以非常高效地计算贝叶斯网络中相对每个结点的偏导。Li 等对贝叶斯网络的最大实例<sup>[7]</sup>、微分<sup>[8]</sup>和敏感性<sup>[9]</sup>分析都进行了比较深入的研究,并把贝叶斯网络融入到产业经济学的研究中,研究宏观经济调控中遇到的相关问题。

时序数据是一类非常重要的数据,目前采用连续时序的贝叶斯网络<sup>[10-11]</sup>来研究这类数据,并揭示它们的规律。近年来,从数据中学习贝叶斯网络结构备受关注,有大量文献<sup>[12-14]</sup>对其进行了研究。Wang 等<sup>[15]</sup>提出了一种新的贝叶斯结构学习算法(Ensemble-Feedback Bayesian Network Struc-

ture Learning, EFBNSL), 该算法首先利用最大信息系数来确定节点间的依赖关系, 以得到初始网络结构, 然后利用 BDe 评分和禁忌搜索算法训练生成若干贝叶斯网络并用邻接矩阵表示。进而利用集成学习策略, 根据得到的若干邻接矩阵计算每条边的权重, 根据集成学习结果利用反馈策略更新初始网络并进入下一次迭代, 经过不断迭代最终确定贝叶斯网络的结构。

## 2.2 因果网的干预

一个最简单的原子干预表示一个变量  $V_i$  被迫取某个固定值  $v_i^k$ , 干预  $\text{do}(v_i^k)$ <sup>[1]</sup> 有两个著名的图解释: 一个是残缺图, 它可以删除所有父结点到干预结点  $V_i$  的边并且设置  $V_i = v_i^k$  和  $P(V_i = v_i^k) = 1$  得到; 另一个是增广网络, 它通过在网络中增加一条假想的干预边  $F_i \rightarrow V_i$  来得到。对多个结点的多干预  $\text{Peal}$ <sup>[1]</sup> 也进行了广泛深入的研究, 并取得了一些很有价值的结论。

我们知道, 在因果网中任何一个结点  $V_i$  可能有很多值, 对结点  $V_i$  进行完全原子干预<sup>[2]</sup> 表示对  $V_i$  中所有值的干预, 也就是对  $V_i$  的每一个值都进行原子干预。一个完全原子干预<sup>[2]</sup> 由  $\text{do}(v_i^1), \dots, \text{do}(v_i^{|V_i|})$  组成, 可以简记为  $\text{do}(V_i)$ 。Qin<sup>[2]</sup> 提出了一个偏原子干预, 它表示一个结点上的一个原子干预和多原子干预; 完全原子干预也可以用偏原子干预来描述。同时提出了一个偏干预表 (Parital Intervention Table, PIT) 来编码对结点的偏原子干预。PIT 中的任何一个元组都具有下面的干预概率: 如果对  $v_i^k$  进行干预, 则  $P'(v_i^k | \mathbf{u}_i) = 1$ ; 否则  $P'(v_i^k | \mathbf{u}_i) = 0$ 。

一个多原子干预的 PIT 表, 是结点的多个原子干预的 PIT 表的并。一个干预的 PIT 表示就是在一个结点把其 CPT 替换成 PIT, Pearl<sup>[1]</sup> 发现如果一个干预结点是无效结点, 那么证据的概率分布是独立于干预结点的, 即干预是无效的, 即  $P(\mathbf{e} | \text{do}(V_i)) = P(\mathbf{e})$ 。例如, 在图 1 的因果网中, 有  $P(c | \text{do}(B)) = P(c)$ 。因此, 每一个完全原子干预有下面的因果效果: 如果  $V_i$  不是一个无效结点, 则  $P(\mathbf{e} | \text{do}(V_i)) = \sum_{v \setminus \{v_i, \mathbf{e}\}} P(v_j | \mathbf{u}_j)$ ; 否则  $P(\mathbf{e} | \text{do}(V_i)) = P(\mathbf{e})$ 。针对因果网最大可能解释的干预<sup>[16]</sup> 问题, Li 等进行了深入研究; 对贝叶斯网络分类问题的基于关系的反事实解释<sup>[17]</sup> 是目前研究的热点问题。

## 3 一个高效的完全原子干预算法

本文在因果网中采用 Park-Darwiche 结构和 PIT 表, 定理 1 提出了一种方法来计算完全原子干预。

**定理 1** 在因果网中构建一个联合树, 对聚簇  $C_l$  中结点  $V_i$  的完全原子干预的计算式如下: 如果  $V_i$  是无效结点, 则  $P(\mathbf{e} | \text{do}(V_i)) = \sum_{(C_l, \mathbf{e}) \setminus \mathbf{e}} \Psi_l / P(v_i | \mathbf{u}_i)$ ; 否则  $P(\mathbf{e} | \text{do}(V_i)) = \sum_{(C_l, \mathbf{e}) \setminus \{v_i, \mathbf{e}\}} \Psi_l / P(v_i | \mathbf{u}_i)$ 。这里,  $\Psi_l = \prod_j M_{jl} \prod_{k=1}^{|C_l|} P(v_k | \mathbf{u}_k)$ ,  $M_{jl}$  表示从聚簇  $C_j$  传到聚簇  $C_l$  的消息。

证明: (1) 如果  $V_i$  是无效结点, 则采用 Park-Darwiche 结构计算完全原子干预, 计算式如下:

$$\begin{aligned} \alpha &= P(\mathbf{e} | \text{do}(V_i)) \\ &= P(\mathbf{e}) \end{aligned}$$

$$\begin{aligned} &= \sum_{(C_l, \mathbf{e}) \setminus \mathbf{e}} P'(v_i | \mathbf{u}_i) \prod_j M_{jl} \prod_{k=1, k \neq i}^{|C_l|} P(v_k | \mathbf{u}_k) \\ &= \sum_{(C_l, \mathbf{e}) \setminus \mathbf{e}} P(v_i | \mathbf{u}_i) \prod_j M_{jl} \prod_{k=1, k \neq i}^{|C_l|} P(v_k | \mathbf{u}_k) / P(v_i | \mathbf{u}_i) \\ &= \sum_{(C_l, \mathbf{e}) \setminus \mathbf{e}} \prod_j M_{jl} \prod_{k=1}^{|C_l|} P(v_k | \mathbf{u}_k) / P(v_i | \mathbf{u}_i) \\ &= \sum_{(C_l, \mathbf{e}) \setminus \mathbf{e}} \Psi_l / P(v_i | \mathbf{u}_i) \end{aligned}$$

(2) 如果  $V_i$  是非无效结点, 我们采用 Park-Darwiche 结构计算其完全原子干预, 计算式如下:

$$\begin{aligned} \alpha &= P(\mathbf{e} | \text{do}(V_i)) \\ &= \sum_{v \setminus \{v_i, \mathbf{e}\}} P'(v_i | \mathbf{u}_i) \prod_j M_{jl} \prod_{k=1, k \neq i}^{|C_l|} P(v_k | \mathbf{u}_k) \\ &= \sum_{v \setminus \{v_i, \mathbf{e}\}} P(v_i | \mathbf{u}_i) \prod_j M_{jl} \prod_{k=1, k \neq i}^{|C_l|} P(v_k | \mathbf{u}_k) / P(v_i | \mathbf{u}_i) \\ &= \sum_{(C_l, \mathbf{e}) \setminus \{v_i, \mathbf{e}\}} \prod_j M_{jl} \prod_{k=1}^{|C_l|} P(v_k | \mathbf{u}_k) / P(v_i | \mathbf{u}_i) \\ &= \sum_{(C_l, \mathbf{e}) \setminus \{v_i, \mathbf{e}\}} \Psi_l / P(v_i | \mathbf{u}_i) \end{aligned}$$

上文的证明中  $v$  表示  $\{C_l, \mathbf{e}\}$ , 定理得证。

根据定理 1 的证明可以看出, 对无效结点的干预不会影响证据结点的概率。我们可以根据文献[3]中提出的方法递归地删除无效结点。而无效结点就是非证据的叶结点, 当删除一个无效结点时就需要删除它的所有入边, 因此可能会出现一些新的无效结点; 但它们也要被递归地删除掉, 直到所有的无效结点都被删除。需要说明的是, 剪枝是递归地删除所有无效结点, 绝不会删除其他结点, 剪枝后的因果网的规模可能比原因果网小很多。根据定理 1, 我们提出了一种优化的联合树算法, 名为 OFDo, 它计算了因果网中对每个结点进行干预时证据的概率值。该算法采用的是 Park-Darwiche 结构, 具体算法如算法 1 所示。

### 算法 1 OFDo 算法

输入: CBN,  $\mathbf{e}$

输出: 对所有结点的完全原子干预

1. 确定  $\mathbf{e}$  涉及的所有证据结点;
2. 在因果网中递归地剪枝掉所有无效结点;
3. 编辑成联合树;
4. 两阶段传递消息;
5. 在任意一个聚簇中计算  $P(\mathbf{e})$ ;
6. for(每一个聚簇  $C_l$ )
7. 从它的所有邻居接受消息;
8. 在各个聚簇中选择一个干预序  $\pi_l$ ;
9. 假设  $\Psi_l = \prod_j M_{jl} \prod_{k=1}^{|C_l|} P(v_k | \mathbf{u}_k)$ ;
10. for(每一个结点  $V_i \in \pi_l$ )
11.  $P(\mathbf{e} | \text{do}(V_i)) = \sum_{(C_l, \mathbf{e}) \setminus \{v_i, \mathbf{e}\}} \Psi_l / P(v_i | \mathbf{u}_i)$

采用算法 1, 我们首先在任意一个聚簇中计算  $P(\mathbf{e})$ , 它等于对任意无效结点的干预值。文献[3]给出了删除无效结点的算法, 并且证明了删除无效结点对最终结果没有影响。算法 1 借鉴了文献[3]中删除无效结点的算法, 因此它是求完全原子干预的精确算法。因为是从非证据的叶结点开始剪枝, 所以在递归剪枝的过程中只需要判断它是否为非证据结点和叶节点, 因此剪枝算法的时间复杂度为  $O(n)$ , 这里  $n$  表示因果网中结点的个数。算法 1 的时间复杂度为  $O(n_0 \exp(\omega_0) + n_0 k_0)$ , 这里  $n_0$  表示剪枝后因果网中的变量的个数,  $\omega_0$  表示

因果网中树的宽度,  $k_0$  是因果网的结点中最大 CPT 中元组的个数。

例 1 假设在图 1 中的因果网中, 证据为  $c=1$ , 我们采用算法 1 计算对所有结点的完全原子干预。在剪枝完后, 因果网只剩下两个结点 A 和 C。对所有非证据结点的完全原子干预的计算式如下:

$$\begin{aligned} P(c=1|\text{do}(A)) &= P(c=1|a) \\ P(c=1|\text{do}(B)) &= P(c=1|\text{do}(D)) = P(c=1|\text{do}(E)) \\ &= \sum_a P(a)P(c=1|a) \end{aligned}$$

#### 4 因果网中的多干预

本文发现因果网中的多干预具有可交换的特性, 定理 2 证明了该特性。

**定理 2** 在因果网中, 我们有:

$$P(\mathbf{e}|\text{do}(V_i), \text{do}(V_j)) = P(\mathbf{e}|\text{do}(V_j), \text{do}(V_i))$$

证明: 我们分以下 3 种情况来证明该定理。

(1)  $V_i$  和  $V_j$  都是无效结点, 该证明过程如下:

$$P(\mathbf{e}|\text{do}(V_i), \text{do}(V_j)) = P(\mathbf{e}|\text{do}(V_j)) = P(\mathbf{e})$$

可以进一步证明:

$$P(\mathbf{e}|\text{do}(V_j), \text{do}(V_i)) = P(\mathbf{e}|\text{do}(V_i)) = P(\mathbf{e})$$

因此, 有  $P(\mathbf{e}|\text{do}(V_i), \text{do}(V_j)) = P(\mathbf{e}|\text{do}(V_j), \text{do}(V_i))$ 。

(2)  $V_i$  或者  $V_j$  是无效结点, 假如  $V_i$  是无效结点, 证明过程如下:

$$\begin{aligned} \alpha &= P(\mathbf{e}|\text{do}(V_i), \text{do}(V_j)) \\ &= P(\mathbf{e}|\text{do}(V_j)) \\ &= \sum_{v \setminus \{v_j, \mathbf{e}\}} P'(v_j | \mathbf{u}_j) \prod_{k \neq i, j} P(v_k | \mathbf{u}_k) \end{aligned}$$

接着, 我们计算  $P(\mathbf{e}|\text{do}(V_j), \text{do}(V_i))$  并且首先干预结点  $V_j$ ; 再采用定理 1, 在因果网中采用 PIT  $P'(v_j | \mathbf{u}_j)$  替换 CPT  $P(v_j | \mathbf{u}_j)$ , 并进行如下推理:

$$P(\mathbf{e}|\text{do}(V_j)) = \sum_{v \setminus \{v_j, \mathbf{e}\}} P'(v_j | \mathbf{u}_j) \prod_{k \neq j} P(v_k | \mathbf{u}_k)$$

我们进一步对结点  $V_i$  进行干预。因为  $V_i$  是一个无效结点, 所以证据的概率分布是独立于结点  $V_i$  的。继续推理得到:

$$P(\mathbf{e}|\text{do}(V_i), \text{do}(V_j)) = \sum_{v \setminus \{v_j, \mathbf{e}\}} P'(v_j | \mathbf{u}_j) \prod_{k \neq i, j} P(v_k | \mathbf{u}_k)$$

因此, 我们有  $P(\mathbf{e}|\text{do}(V_i), \text{do}(V_j)) = P(\mathbf{e}|\text{do}(V_j), \text{do}(V_i))$ 。如果  $V_j$  是一个无效结点, 采用类似的方法也能得到  $P(\mathbf{e}|\text{do}(V_i), \text{do}(V_j)) = P(\mathbf{e}|\text{do}(V_j), \text{do}(V_i))$ 。

(3)  $V_i$  和  $V_j$  都不是无效结点。我们计算  $P(\mathbf{e}|\text{do}(V_i), \text{do}(V_j))$ , 首先对结点  $V_i$  进行干预。采用定理 1, 在因果网中采用 PIT  $P'(v_i | \mathbf{u}_i)$  来取代 CPT  $P(v_i | \mathbf{u}_i)$ , 并得到如下计算式:

$$P(\mathbf{e}|\text{do}(V_i)) = \sum_{v \setminus \{v_i, \mathbf{e}\}} P'(v_i | \mathbf{u}_i) \prod_{k \neq i} P(v_k | \mathbf{u}_k)$$

接着, 对结点  $V_j$  进行干预。然后采用定理 1, 在因果网中采用 PIT  $P'(v_j | \mathbf{u}_j)$  来取代 CPT  $P(v_j | \mathbf{u}_j)$ , 并得到如下计算式:

$$P(\mathbf{e}|\text{do}(V_i), \text{do}(V_j)) = \sum_{v \setminus \{v_i, v_j, \mathbf{e}\}} \prod_{l=i, j} P'(v_l | \mathbf{u}_l) \prod_{k \neq i, j} P(v_k | \mathbf{u}_k)$$

下一步计算  $P(\mathbf{e}|\text{do}(V_j), \text{do}(V_i))$ , 首先对结点  $V_j$  进行干预。采用定理 1, 在因果网中我们采用 PIT  $P'(v_j | \mathbf{u}_j)$  来取代 CPT  $P(v_j | \mathbf{u}_j)$ , 并得到如下计算式:

$$P(\mathbf{e}|\text{do}(V_j)) = \sum_{v \setminus \{v_j, \mathbf{e}\}} P'(v_j | \mathbf{u}_j) \prod_{k \neq j} P(v_k | \mathbf{u}_k)$$

接着, 对结点  $V_i$  进行干预。再采用定理 1, 在因果网中我们采用 PIT  $P'(v_i | \mathbf{u}_i)$  来取代 CPT  $P(v_i | \mathbf{u}_i)$ , 并得到如下计算式:

$$P(\mathbf{e}|\text{do}(V_j), \text{do}(V_i)) = \sum_{v \setminus \{v_i, v_j, \mathbf{e}\}} \prod_{l=i, j} P'(v_l | \mathbf{u}_l) \prod_{k \neq i, j} P(v_k | \mathbf{u}_k)$$

因此, 我们可以得到  $P(\mathbf{e}|\text{do}(V_i), \text{do}(V_j)) = P(\mathbf{e}|\text{do}(V_j), \text{do}(V_i))$ , 则定理得证。

因为对同一结点的多原子干预是相互独立的, 所以它们也具有可交换特性, 因此有  $P(\mathbf{e}|\text{do}(v_i^l), \text{do}(v_i^l)) = P(\mathbf{e}|\text{do}(v_i^l), \text{do}(v_i^l))$ 。在干预的可交换性基础上, 我们将证明多干预定理 3。

**定理 3** 在一个具有  $n$  个结点的因果网中, 我们计算干预  $P(\mathbf{e}|\text{do}(V_1), \dots, \text{do}(V_m))$ , 这里  $m \leq n$ 。我们考虑以下 3 种情况:

(1) 假如所有的干预结点  $V_i (i=1, \dots, m)$  都是无效结点, 则有  $P(\mathbf{e}|\text{do}(V_1), \dots, \text{do}(V_m)) = P(\mathbf{e})$ 。

(2) 假如有  $k (k < m)$  个干预结点是无效结点, 则有  $P(\mathbf{e}|\text{do}(V_1), \dots, \text{do}(V_m)) = \sum_{v \setminus \{v_{k+1}, \dots, v_m, \mathbf{e}\}} \prod_{l=m+1}^m P(v_l | \mathbf{u}_l) \prod_{l=k+1}^m P'(v_l | \mathbf{u}_l)$ 。

(3) 如果所有的干预结点  $V_i (i=1, \dots, m)$  都是有效的, 则有  $P(\mathbf{e}|\text{do}(V_1), \dots, \text{do}(V_m)) = \sum_{v \setminus \{v_1, \dots, v_m, \mathbf{e}\}} \prod_{l=m+1}^n P(v_l | \mathbf{u}_l) \prod_{l=1}^m P'(v_l | \mathbf{u}_l)$ 。

证明: 本文采用如下方式证明定理 3 中的 3 种情况。

(1) 所有的干预结点  $V_i (i=1, \dots, m)$  都是无效的, 证明过程如下:

$$\begin{aligned} \alpha &= P(\mathbf{e}|\text{do}(V_1), \dots, \text{do}(V_m)) \\ &= P(\mathbf{e}|\text{do}(V_2), \dots, \text{do}(V_m)) \\ &= \dots \\ &= P(\mathbf{e}) \end{aligned}$$

(2)  $k$  个干预结点是无效的。我们首先采用定理 2 对这些结点进行重新排序, 使  $\{V_1, \dots, V_k\}$  是  $k$  个无效结点。证明过程如下:

$$\begin{aligned} \alpha &= P(\mathbf{e}|\text{do}(V_1), \dots, \text{do}(V_m)) \\ &= P(\mathbf{e}|\text{do}(V_2), \dots, \text{do}(V_m)) \\ &= \dots \\ &= P(\mathbf{e}|\text{do}(V_{k+1}), \dots, \text{do}(V_m)) \\ &= \sum_{v \setminus \{v_{k+1}, \dots, v_m, \mathbf{e}\}} \prod_{l=m+1}^m P(v_l | \mathbf{u}_l) \prod_{l=k+1}^m P'(v_l | \mathbf{u}_l) \end{aligned}$$

(3) 所有结点  $V_i$  都是有效结点, 证明过程如下:

$$\begin{aligned} \alpha &= P(\mathbf{e}|\text{do}(V_1), \dots, \text{do}(V_m)) \\ &= \sum_{v \setminus \{v_{k+1}, \dots, v_m, \mathbf{e}\}} \prod_{l=m+1}^m P(v_l | \mathbf{u}_l) \\ &= \sum_{v \setminus \{v_{k+1}, \dots, v_m, \mathbf{e}\}} \prod_{l=m+1}^m P(v_l | \mathbf{u}_l) \prod_{l=1}^m P'(v_l | \mathbf{u}_l) \end{aligned}$$

因此, 定理得证。

假如一个剪枝后的因果网有  $m$  个非证据结点和  $l$  个证据结点, 我们不对证据结点进行干预。假如我们对所有的非证据结点进行干预  $V_i (i=1, \dots, m)$ , 则有如下的干预效果:

$$P(\mathbf{e}|\text{do}(V_1), \dots, \text{do}(V_m)) = \prod_{k=1}^m P'(v_k | \mathbf{u}_k) \prod_{i=1}^l P(v_i | \mathbf{u}_i)$$

假如在图1的因果网中,我们对结点B和C进行干预,则干预的效果为:

$$\begin{aligned} \alpha &= P(d=0, e=1 | do(B), do(C)) \\ &= \sum_a P(a) P'(b|a) P'(c|a) P(d=0 | b, c) P(e=1 | d) \\ &= \sum_a P(a) P(d=0 | b, c) P(e=1 | d) \\ &= P(d=0 | b, c) P(e=1 | d) \end{aligned}$$

$P(d=0, e=1 | do(B), do(C))$ 的结果如表1所列。

表1  $P(d=0, e=1 | do(B), do(C))$ 的值

Table 1 Result of  $P(d=0, e=1 | do(B), do(C))$

B	C	D	E	p
0	0	0	1	0.5808
0	1	0	1	0.0902
1	0	0	1	0.033
1	1	0	1	0.0615

由表1可知,当 $B=0$ 并且 $C=0$ 时,干预的效果最好。因此,多干预的一个重要应用是找到最好的干预组合。

## 5 实验

在BNJ<sup>1)</sup>的基础上,我们实现了OFDo算法;另外,我们还采用Java实现了Pearl的Do算法<sup>[1]</sup>和文献[2]中的FDo算法。我们的实验环境是一个英特尔3.4GHz 8核i7-6700 CPU,16GB内存和Linux操作系统。我们对每个因果网的所有非证据结点进行全原子干预,实验中采用的因果网是从相关标准<sup>[16]2)</sup>中找出的。

因为FDo算法和Do算法都是采用基于联合树的Park-Darwiche结构实现的,所以首先为它们编辑联合树,并采用CP表示FDo算法和Do算法的编辑阶段。对于OFDo算法,我们采用CP-O表示它的剪枝和编辑阶段,也就是说CP-O表示的是对因果网进行剪枝以及把剪枝后的因果网编辑成联合树所需的时间。接着分别计算对每个结点进行完全原子干预时证据的概率,也就是干预的效果。我们采用OFDo-I, FDo-I和Do-I来分别表示在下行阶段对每个结点计算完全原子干预时分别采用算法OFDo, FDo和Do的性能。从表2中可以发现,相比Do, FDo在所有的因果网中的干预效果都好,这是因为FDo算法在对各个结点计算完全原子干预时,只需要一次概率推理;而Do算法在对每个结点计算原子干预时采用一次概率推理,也就是说如果一个结点有 $k$ 个值,则需要进行 $k$ 次概率推理来计算该结点的完全原子干预。因为它们都采用Park-Darwiche结构,所有传递到一个聚簇的消息只需要计算一次。

从表2中还发现,CP-O比CP的性能更好。这是因为剪枝后的因果网比原因果网小很多,因此编辑原因果网需要的时间比编辑剪枝后的因果网需要的时间长。因此,一个好的选择是先剪枝一个因果网,然后再编辑它,这比直接编辑原因果网要好。我们进一步地发现,OFDo-I的性能比FDo-I的性能好很多,主要有如下两个原因:1)OFDo-I对所有的无效结点只计算一次干预的效果,而FDo-I必须对所有的无效结点逐一计算干预的效果;2)原因果网比剪枝后的因果网大很多。

表2 OFDo和FDo以及Do的性能对比

Table 2 Performance of OFDo vs. FDo and Do

(单位:ms)

Networks	CP	CP-O	FDo-I	Do-I	OFDo-I
alarm	2.5	2.2	1	1.1	0.6
bm_05_01	130.3	102.6	9.9	11.9	2
bm_05_02	210.8	175.9	13.2	14.3	3.4
bm_05_03	204	164.8	12.2	13.4	3.6
diabetes	14.3	11.5	11.4	17.1	2
fs-04	6.1	5.3	0.8	0.9	0.6
hailfinder	3.8	3.5	1.1	1.4	0.8
mm_3_8_3	185	166.6	7.4	8.1	0.8
munin1	146	124.5	17.1	28.3	3.8
munin2	22	16.3	12.2	15.9	3.6
munin3	21.5	17	12.5	15	3.1
munin4	26.2	19.8	15.9	22.3	3.4
pathfinder	18.1	13.4	6.3	8.2	3.5
pigs	10.1	8.4	3.3	4	1.4
Students-3_2	22	16.9	4.3	4.7	0.8
water	7.9	6.6	2.6	3.6	1.2

**结束语** 贝叶斯网络在人工智能、信号处理、信息检索、传感器、普适计算、自然语言处理和计算机视觉等领域是一个非常通用的工具。因果网是基于贝叶斯网络的另外一种概率图模型,它可以用来表示对外界的反应。因为一个无效结点对边际概率没有贡献,所以我们应该先剪枝一个因果网,然后在剪枝后的因果网上计算干预的效果。接着,我们设计了一种优化算法来对因果网中的各个结点进行完全原子干预。最后,我们证明多干预具有可交换性。政府在对宏观经济进行干预时,需要忽略那些对预期效果没有影响的因素,也就是需要删除那些类似于因果网中的无效结点。未来我们将采用本文提出的技术在统计分析中高效地执行干预效果的计算。

## 参考文献

- [1] PEARL J. Causality: models, reasoning, and inference[M]. Cambridge University Press, 2009.
- [2] QIN B. Differential semantics of intervention in Bayesian networks[C]//Proc. of the 24th International Joint Conference on Artificial Intelligence. 2015:710-716.
- [3] SHACHTE R. Evaluating influence diagrams[J]. Operations Research, 1986, 34(6):871-882.
- [4] PARK J, DARWICHE A. Morphing the hugin and shenoy-shafer architectures[C]//7th European Conference Symbolic and Quantitative Approaches to Reasoning with Uncertainty. 2003:149-160.
- [5] SHENOY P, SHAFER G. Propagating belief functions with local computations[J]. IEEE Expert, 1986, 1(3):43-52.
- [6] JENSEN F, ANDERSEN S. Approximations in Bayesian belief universes for knowledge based systems[C]//Proc. of the 6th Conference on Uncertainty Artificial Intelligence. 1990:162-169.
- [7] LI C, QIN B. New algorithms for computing max-product instantiations[J]. Application Research of Computers, 2015, 32(6):1711-1715.

<sup>1)</sup> <http://sourceforge.net/projects/bnj/>

<sup>2)</sup> <http://reasoning.cs.ucla.edu/ace>

- [8] LI C. Jointree algorithm for the differentiation of the MPE problem[J]. Journal of Chinese Computer Systems, 2016, 37(10): 2306-2311.
- [9] QIN B, WANG Q Y, LI C. An effective strategy for sensitivity analysis of Bayesian networks[J]. Journal of Chinese Computer Systems, 2016, 37(4): 732-737.
- [10] LINZNER D, SCHMIDT M, KOEPL H. Scalable Structure Learning of Continuous-Time Bayesian Networks from Incomplete Data[C]//Proc. of the 33rd Advances in Neural Information Processing Systems. 2019; 3741-3751.
- [11] BHATTACHARJYA D, SHANMUGAM K, GAO T, et al. Event-Driven Continuous Time Bayesian Networks[C]//Proc. of the 34th AAAI Conference on Artificial Intelligence. 2020; 3259-3266.
- [12] GRUTTEMEIER N, KOMUSIEWICZ C. Learning Bayesian Networks Under Sparsity Constraints: A Parameterized Complexity Analysis[C]//Proc. of the 29th International Joint Conference on Artificial Intelligence. 2020; 4245-4251.
- [13] CHEN C, YUAN C. Learning Diverse Bayesian Networks[C]//Proc. of the 33th AAAI Conference on Artificial Intelligence. 2019; 7793-7800.
- [14] LIAO Z, SHARAM C, CUSSENS J, et al. Finding All Bayesian Network Structures within a Factor of Optimal[C]//Proc. of the 33th AAAI Conference on Artificial Intelligence. 2019; 7892-7899.
- [15] WANG S H, QIN B. Bayesian Network Structure Learning by Ensemble Learning and Feedback Strategy[J]. Chinese Journal of Computers, 2021, 44(6): 1051-1063.
- [16] LI C, QIN B. Efficient Computation of MPE in Causal Bayesian Networks[J]. Computer Science, 2021, 48(4): 14-19.
- [17] ALBINI E, RAGO A, BARONI P, et al. Relation-Based Counterfactual Explanations for Bayesian Network Classifiers[C]//Proc. of the 29th International Joint Conference on Artificial Intelligence. 2020; 451-457.



**LI Chao**, born in 1976, Ph.D, professor. Her main research interests include statistical machine learning and business intelligence.



**QIN Biao**, born in 1972, Ph.D, vice professor, Ph.D supervisor, is a member of China Computer Federation. His main research interests include causal Bayesian network and machine learning.

(责任编辑:喻黎)