

高效计算因果网中的最大可能解释



李超¹ 覃飙²

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

(chaoli@cupl.edu.cn)

摘要 在因果网中,高效计算的最大可能解释(Most Probable Explanations,MPE)是一个关键问题。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。文中通过比较干预和微分的语义,揭示了 MPE 完全原子干预的微分语义。根据微分语义,因果网中原子干预 MPE 实例的计算可以归约为贝叶斯网络中的 MPE 实例的计算。接着,提出了一个联合树算法(Best JoinTree,BJT),它通过在因果网中只构建一个联合树来计算最好的原子干预,原子干预的结果包含一个 BMPE(Best MPE)概率和它对应的实例。其中,BMPE 概率是对 MPE 所有结点分别进行原子干预后得到的最高概率。BJT 可以采用干预的效果来计算对应贝叶斯网络的 MPE 概率和 MPE 实例。最后,实验证实了绝大多数因果网在计算最好原子干预时,BJT 的速度比目前最好的算法快了超过 10 倍。

关键词 因果网;贝叶斯网络;干预;微分 MPE;MPE 实例

中图分类号 TP311

Efficient Computation of MPE in Causal Bayesian Networks

LI Chao¹ and QIN Biao²

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

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

Abstract This paper investigates the efficient computation of most probable explanations(MPE) in causal Bayesian networks(CBNs). From the perspective of a directed acyclic graph, researchers find that every CBN has a corresponding Bayesian network. By comparing the semantics of intervention with that of differentiation, this paper reveals the differential semantics of a full atomic intervention on MPE. Using the differential semantics, it reduces the computation of an MPE instantiation for an atomic intervention on a CBN to that of an MPE instantiation in the corresponding Bayesian network. Next, it proposes a jointtree algorithm called BJT to compute the best atomic intervention, which includes the BMPE probability and a corresponding instantiation, by building only one jointtree for a CBN. The BMPE probability is the highest probability of all atomic interventions on MPE in a CBN. BJT can use the causal effect to compute the MPE probability and an MPE instantiation for the corresponding Bayesian network. Finally, experimental results show that BJT performs more than one order of magnitude faster than the-state-of-the-art algorithm for computing the best atomic intervention in most CBNs.

Keywords Causal Bayesian networks, Bayesian networks, Intervention, Differentiation of MPE, MPE instantiation

1 引言

因果网(Causal Bayesian Networks,CBN)^[1]采用有向无环图(Directed acyclic graph,DAG)来描述因果信息和传统的条件依赖,通过概率演算来计算因果效果,其关键点是 do-算子,它把因果查询归约为概率查询。一个原子干预可以表示

为 $do(V_i = v_i^k)$ 或者简计为 $do(v_i^k)$ 。do-算子可以从代数或图形上进行解释。贝叶斯网络是另一个重要的 DAG,它也采用概率演算来计算信任的程度。如果结合干预操作和贝叶斯网络 DAG 的不变性假设,就能得到一个新的因果网系统。因此从 DAG 的角度出发,因果网有对应的贝叶斯网络。这样我们可以把因果网中干预效果的计算转换为贝叶斯网络中概

投稿日期:2020-05-28 返修日期:2020-08-12 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(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 Natural Science Foundation of China(61772534), Key Projects of Philosophy and Social Sciences Research of Ministry of Education(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:覃飙(qinbiao@ruc.edu.cn)

率的计算。最大可能解释(Most Probable Explanation, MPE)是贝叶斯网络中一类重要的查询, MPE问题可以记为 $MPE(e)$, 它计算证据 e 的最大概率以及相应的 MPE 实例, 这个最大概率称为 MPE 概率, 它记为 $MPE_p(e)$ 。Li 等^[2] 提出了计算 MPE 实例的新算法。

在因果网中, 对 MPE 的一个原子干预可以记为 $MPE(e|do(v_i^k))$, 它计算在给定原子干预 $do(v_i^k)$ 情况下的最大概率和对应的各个网络变量的实例。Nielsen 等^[3] 对因果网中 MPE 的干预进行了研究, 提出了采用解释树寻找近似的 MPE 实例。本文高效地计算最好的原子干预, 这在因果网中包括 BMPE 概率和对应的实例, 前者是因果网中对所有结点分别进行 MPE 类型原子干预得到的最大概率。进一步, 本文研究了因果网对 MPE 原子干预的精确 MPE 实例。这个研究有很多现实的应用, 例如当对经济进行宏观调控时, 政府会关注哪个产业能够最大化这种政策干预的效果。然而, 采用目前的最新技术^[2] 来找到这个最好的原子干预, 需要构建大量的解释树, 其时间复杂度过高。

一个完全的原子干预^[4] 由 $do(v_1^k), \dots, do(v_n^k)$ 组成, 可以简记为 $do(V_i)$ 。Qin^[4] 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, 同时提出了一个偏干预表(Parital Intervention Table, PIT)来编码偏原子干预。因此 Qin 在计算干预效果时把条件概率表(Conditional Probability Table, CPT)替换为 PIT。在 PIT 的基础上, Qin 揭示了和积问题的完全原子干预问题的微分语义。然而我们的研究表明, 在计算干预效果时不用把 CPT 替换为 PIT, 并且不需要物理地生成各个 PIT, 它仅仅用来从逻辑上解释 do-算子。

Park 等^[5] 揭示了贝叶斯网络联合树的微分语义; 另外, 贝叶斯网络的微分问题在其敏感性分析^[6] 中得到广泛应用。在此基础上, 本文揭示了因果网中对 MPE 干预的微分语义。由于一个因果网具有一个对应的贝叶斯网络, 本文提出了一个基于联合树的 BJT 算法, 它通过仅构造一个联合树来计算最好原子干预。通过采用 MPE 完全原子干预的微分语义, BJT 把因果网中计算 MPE 原子干预实例的计算问题归结为对应贝叶斯网络中 MPE 实例的计算问题。在计算对某个结点 V_i 的完全原子干预后, BJT 用该完全原子干预来计算对应贝叶斯网络中 V_i 及其父结点的实例; 同时 BJT 可以采用数据特性来优化在对应贝叶斯网络中 MPE 实例的计算。当然, 如果把 $MPE(e|do(V))$ 换成 $P(e|do(V))$, BJT 就能够用来计算因果网中的原子干预、完全原子干预和偏原子干预。

本文的主要贡献在于: 1) 揭示了因果网中 MPE 干预的微分语义; 2) 根据该微分语义, 把因果网中原子干预的 MPE 实例的计算问题归结为对应的贝叶斯网络中 MPE 实例的计算; 3) 提出了一个联合树算法 BJT 来计算最好原子干预。

2 贝叶斯网络和因果网的基本知识

2.1 贝叶斯网络的计算和微分

一个贝叶斯网络采用一个 DAG 来表示一组随机变量和它们的条件依赖, 每个变量采用贝叶斯网络中的一个结点表示, 每个结点有一个条件概率表(CPT)与之相关联。在给定父结点的情况下, 每个变量条件独立于图中它的所有非孩子结点。贝叶斯网络中有 3 个非常重要的规则: 链式规则、全概率规则和贝叶斯规则。链式规则允许从简单部件构造复杂模

型; 在链式规则下, 有 $P(\gamma) = \prod P(x|u)$, 这里 γ 是贝叶斯网络中所有变量 Γ 的一个实例。最近在贝叶斯网络中采用搜索来计算其概率值^[7-8] 的方法得到了较多关注。

在联合树算法中, 由于计算微分时 Shenoy-Shafer^[9] 的联合树比 Hug 联合树^[10] 包含更多的信息^[5], 因此本文采用 Shenoy-Shafer 的联合树, 其处理过程如下。首先, 在网络参数初始化时, 证据 e 被联合树接受。接着, 一个簇被选为根簇, 并且消息从两个方向进行传播, 上行阶段把消息向根簇传播, 下行阶段根簇把消息传给其他簇。在联合树中从簇 C_i 到簇 C_j 的分离器是一组变量 $S_{ij} = vars(f_i) \cap vars(f_j)$, 这里 $vars(f_i)$ 表示出现在因子和簇 C_i 的消息中的变量。簇 C_i 发送消息 $M_{ij} = \sum_{C_i \setminus S_{ij}} \Phi_i \prod_{k \neq j} M_k$, 给簇 C_j 的条件是 C_i 从其他所有邻接簇 C_k 接受到消息后才把消息传递给 C_j , 这里 Φ_i 是簇 C_i 各个因子的乘积。一旦消息传播结束, 则可以计算证据 e 和任何簇中变量的联合概率分布: $P(C_i, e) = \Phi_i \prod_k M_k$ 。

在 MPE 中, 一个实例被称为 MPE 实例, 其对应的概率被称为 MPE 概率, MPE 概率可以定义为 $MPE_p(e) = \max_m P(m, e)$, 这里 m 是变量集合中所有非证据结点的实例。因此有 $\gamma = mUe$, 此时可能有许多实例 m 都满足这个最大概率, 每个这样的实例都是 MPE 实例, 这一组实例可以定义为 $MPE(e) = \arg \max_m P(m, e)$ 。

在贝叶斯网络中, 如果 $P(v_i | u_i) \neq 0$, MPE 的微分^[11] 被证明可以为如下等式:

$$\frac{\partial MPE_p(e)}{\partial P(v_i | u_i)} = MPE_p(v_i, u_i, e) / P(v_i | u_i) = \max_{\lambda(v_i, u_i, e)} P(v_i | u_i) \quad (1)$$

2.2 因果网的干预

因果网是另一个重要的 DAG, 它编码了许多不变形假设, 这样可以计算干预的效果, 并且贝叶斯网络包含在因果网中。一个原子干预表示一个变量 V_i 被迫取某个固定值 v_i^k 。

$do(v_i^k)$ ^[11] 有两个著名的图解释: 1) 残缺图, 它可以从删除所有父结点到干预结点 V_i 的边得到; 2) 增广网络, 它通过在网络中增加一条假想的干预边 $F_i \rightarrow V_i$ 得到。图 1 给出了具有 5 个结点的贝叶斯网络, 图 2(a) 和图 2(b) 分别表示对图 1 的因果网进行干预 $do(D=1)$ 所产生的残缺图和增广图。

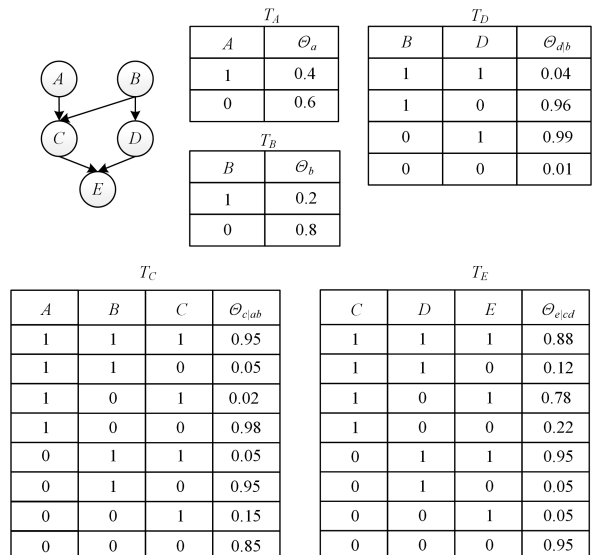
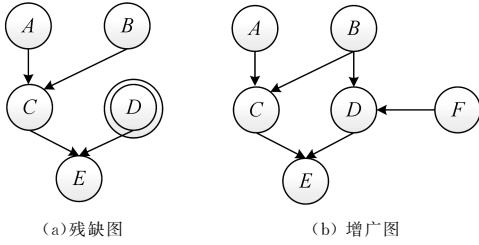


图 1 具有 5 个结点的贝叶斯网络

Fig. 1 Bayesian network with five nodes

图2 干预为 $D=1$ 时的网络表示Fig. 2 Network representation of action "setting $D=1$ "

根据 Pearl 的描述^[1],若执行干预 $do(v_i^k)$,则干预概率为 $P(v_i^k)=1$,否则 $P(v_i^k)=0$ 。例如,在图1所示的因果网中执行干预 $do(D=1)$,我们设置 $P(D=1)=1$ 和 $P(D=0)=0$ 。Qin^[4]提出了一个偏干预表(PIT)来编码偏原子干预。由此,在因果网中计算干预效果时,可以生成 PIT 并采用它来取代 CPT。

对和积问题的干预^[1,4,12]到现在还是一个非常流行的问题,到目前为止只有一篇文献^[3]研究了因果网中对 MPE 的干预问题,并且它计算的是近似 MPE 概率。这是因为对因果网中 MPE 的干预需要计算 MPE 实例,按照传统算法则需要对每个结点进行干预后构建一个联合树来计算 MPE 实例,这样计算的代价太大。本文只采用一个联合树来计算对各个结点进行干预的 MPE 实例。

3 对 MPE 进行干预

对因果网中的 MPE 进行原子干预,干预点为 V_i 且证据为 e ,则干预的效果记为 $MPE_p(e|do(v_i^k))$,并且有定义: $MPE_p(e|do(v_i^k)) = \max_{m'} P(m', e|do(v_i^k))$,计算得到的概率为在对 MPE 进行干预时的 MPE 概率,相应的实例为 MPE 实例。这里, m' 表示除去证据变量和干预变量之后的其他所有变量;可能有 m' 的大量实例满足最大概率,其中每个实例都是 MPE 实例,这些实例满足如下定义:

$$MPE_p(e|do(v_i^k)) = \operatorname{argmax}_{m'} P(m', e|do(v_i^k)) \quad (2)$$

下面研究对 MPE 完全原子干预的特性,并阐述在联合树中处理因果网中的 MPE。

3.1 完全干预 MPE 的微分语义

因为结点 V_i 的各个值是相互独立的,所以根据残缺图^[1],我们可以首先计算各个原子干预 $do(v_i^k)$ ($k=1, \dots, n$),然后将它们合并得到对结点 V_i 的完全原子干预。有下面两种情况:

1) V_i 是一个非叶结点。对于 MPE 的原子干预我们采用下式计算 MPE 的概率:

$$MPE_p(e|do(v_i^k)) = \max_{\lambda_e} \prod_{j \neq i, V_j = v_k} P(v_j | \mathbf{u}_j)$$

当 $k \neq k'$, 因为 $MPE_p(e|do(v_i^k))$ 一般不等于 $MPE_p(e|do(v_i^{k'}))$,所以在计算 MPE 的完全原子干预时需要计算对 V_i 各个取值的干预,我们保留等式(3)右边的 V_i 来达到这个目的,如式(3)所示:

$$MPE_p(e|do(v_i)) = \max_{\lambda(v_i, e)} \prod P(v_j | \mathbf{u}_j) \quad (3)$$

2) V_i 是一个叶结点。因为 V_i 不可能是其他结点的父

结点,所以有 $MPE_p(e|do(v_i^k)) = MPE_p(e|do(v_i^{k'}))$,这里 $k \neq k'$ 。因此,对叶结点的干预不影响证据的 MPE 值,我们得到:

$$MPE_p(e|do(v_i)) = \max_{\lambda_e} \prod_{j \neq i} P(v_j | \mathbf{u}_j)$$

依据等式(4),我们可以采用对叶结点的原子干预计算该结点的完全原子干预。根据式(3)和式(4),我们发现,单个概率推导可以计算一个结点上对 MPE 的完全原子干预。进一步,假如对结点 V_i 执行完全原子干预,可以观察到 V_i 既没有卷入式(3)也没有卷入式(4)。因此,可以利用该特性计算对 MPE 的完全原子干预,此时既不需要改变网络结构,也不需要生成 PIT,并把 CPT 替换为 PIT^[3]。在完成对所有结点执行 MPE 的完全原子干预后,可以很容易地找到最好原子干预,它的定义如定义1所示。

定义1 因果网中一个原子干预 $do(v_i^k)$,如果它在所有的原子干预中得到证据 e 的 MPE 概率最大,则该原子干预被称为最好原子干预。最好原子干预的概率称为 BMPE 概率,可以表示为 $BMPE_p(e)$,其对应的任意一个实例都可以称作 BMPE 实例。

我们采用下面的例子说明在因果网中如何计算 BMPE 概率。

例1 在图1的因果网中,假如证据为 $D=1$ 。根据图1的初始值采用式(3)和式(4)计算对非证据结点完全原子干预的 MPE 概率,结果展示在图3中。从图3中可以观察到两个现象:1)对叶结点 E 的两个原子干预有相同的效果,都为 0.4039,这种现象与式(4)是一致的;2)最好的原子干预 $do(A=1)$,其对应的 BMPE 概率为 $BMPE_p(D=1) = 0.7374$ 。我们观察到对 $B=1$ 的原子干预为 $MPE_p(D=1|do(B=1)) = 0.0217$,而对 $B=0$ 的原子干预效果为 $MPE_p(D=1|do(B=0)) = 0.4797$,由此可以看出,对同一结点最大和最小的干预效果差距可能是非常大的。因此我们不能利用对某个结点的的一个干预效果 $do(v_i^k)$ 来估计对该结点的其他干预效果 $do(v_i^{k'})$ ($k \neq k'$)。下节将提出联合树算法来计算 BMPE 实例。

$MPE_p(D=1 do(A))$			$MPE_p(D=1 do(B))$		
A	D	p	B	D	p
1	1	0.7374	1	1	0.0217
0	1	0.6395	0	1	0.4797

$MPE_p(D=1 do(C))$			$MPE_p(D=1 do(E))$		
C	D	p	E	D	p
1	1	0.4182	1	1	0.4039
0	1	0.4514	0	1	0.4039

图3 对图1因果网中所有非证据结点进行完全原子干预
Fig. 3 Full atomic interventions on all non-evidence nodes in CBN shown in Fig. 1

如果 V_i 是式(1)中的叶结点,它将不会出现在 $\prod_{j \neq i} P(v_j | \mathbf{u}_j)$ 中,因而能够导出:

$$\partial MPE_p(e) / \partial P(v_i | \mathbf{u}_i) = \max_{\lambda(\mathbf{u}_i, e)} \prod_{j \neq i} P(v_j | \mathbf{u}_j) \quad (5)$$

如果 V_i 是式(1)中的根结点,它没有父结点,则容易得到:

$$\partial MPE_p(\mathbf{e})/\partial P(v_i) = \max_{\lambda \setminus \{v_i, \mathbf{e}\}} \prod_{j \neq i} P(v_j | \mathbf{u}_j) \quad (6)$$

因此,在因果网中,如果计算叶结点的干预,则可通过式(4)和式(5);如果计算根结点的干预则可通过等式(3)和等式(6);如果计算其他结点的干预则可通过等式(3)和等式(1)。那么,可以得到如下对 MPE 完全原子干预的微分语义:

当 V_i 是根结点时, $MPE_p(\mathbf{e} | do(v_i)) = \partial MPE_p(\mathbf{e}) / \partial P(v_i)$; 当 V_i 是非根结点时, $MPE_p(\mathbf{e} | do(v_i)) = \max_{u_i} \partial MPE_p(\mathbf{e}) / \partial P(v_i | \mathbf{u}_i)$

3.2 计算最好原子干预的联合树算法

在 MPE 完全原子干预微分语义的基础上,定理 1 证明了因果网中 $MPE_p(\mathbf{e} | do(v_i^k))$ 实例的计算可以归约为计算对应贝叶斯网络中的 $MPE_p(\mathbf{e}, v_i^k, u_i^k)$, 该定理把因果网和贝叶斯网络联系起来,起到了桥梁的作用;同时,一个因果网有一个对应的贝叶斯网络^[4], 是采用贝叶斯网络中的联合树计算干预的前提和基础。

定理 1 因果网原子干预 $MPE_p(\mathbf{e} | do(v_i^k))$ 和对应贝叶斯网络中计算 MPE 的联系如下:

1) 如果 V_i 是非根结点,则有如下等式:

$$MPE_p(\mathbf{e}, v_i^k, u_i^k) = P(v_i^k | u_i^k) MPE_p(\mathbf{e} | do(v_i^k))$$

2) 如果 V_i 是根结点,则有如下等式:

$$MPE_p(\mathbf{e}, v_i^k) = P(v_i^k) MPE_p(\mathbf{e} | do(v_i^k))$$

这里 V_i 是原子干预结点, U_i 是其父结点, v_i^k 和 u_i^k 分别为它们的实例。

证明:

1) V_i 是非根结点。根据 MPE 完全原子干预的微分语义,有下面的等式:

$$MPE_p(\mathbf{e} | do(v_i^k)) = \max_{u_i} \partial MPE_p(\mathbf{e}) / \partial P(v_i^k | \mathbf{u}_i)$$

设 $\max(\partial MPE_p(\mathbf{e}) / \partial P(v_i^k | \mathbf{u}_i))$ 表示 $\partial MPE_p(\mathbf{e}) / \partial P(v_i^k | \mathbf{u}_i)$ 结果中所有元组的最大概率。根据式(1)可以进一步进行如下推导:

$$\begin{aligned} MPE_p(\mathbf{e} | do(v_i^k)) &= \max(\partial MPE_p(\mathbf{e}) / \partial P(v_i^k | \mathbf{u}_i)) \\ &= \max(MPE_p(v_i^k, \mathbf{u}_i, \mathbf{e}) / P(v_i^k | \mathbf{u}_i)) \end{aligned}$$

将等式两边同时乘以 $P(v_i^k | \mathbf{u}_i)$, 得到如下等式:

$$P(v_i^k | \mathbf{u}_i) MPE_p(\mathbf{e} | do(v_i^k)) = P(v_i^k | \mathbf{u}_i) \max(MPE_p(v_i^k, \mathbf{u}_i, \mathbf{e}) / P(v_i^k | \mathbf{u}_i))$$

因此,我们有:

$$\max(P(v_i^k | \mathbf{u}_i) MPE_p(\mathbf{e} | do(v_i^k))) = \max(MPE_p(v_i^k, \mathbf{u}_i, \mathbf{e}))$$

我们能够很容易地计算实例 $U_i = u_i^k$, 使上面等式的左侧具有最大概率。因此可以得到:

$$MPE_p(\mathbf{e}, v_i^k, u_i^k) = P(v_i^k | u_i^k) MPE_p(\mathbf{e} | do(v_i^k))$$

2) V_i 是根结点。结合式(1)和 MPE 完全原子干预的微分语义,我们进行如下推导:

$$\begin{aligned} MPE_p(\mathbf{e} | do(v_i^k)) &= \partial MPE_p(\mathbf{e}) / \partial P(v_i^k) \\ &= MPE_p(v_i^k, \mathbf{e}) / P(v_i^k) \end{aligned}$$

因此能够得到:

$$MPE_p(\mathbf{e}, v_i^k) = P(v_i^k) MPE_p(\mathbf{e} | do(v_i^k))$$

定理 1 得证。

利用定理 1, 我们得到 $MPE_p(\mathbf{e} | do(v_i^k))$ 和 $MPE_p(\mathbf{e}, v_i^k, u_i^k)$ 具有相同的实例。我们可以进一步归约 $MPE_p(\mathbf{e}, v_i^k, u_i^k)$ 实例的计算到 $MPE_p(\mathbf{e})$ 实例的计算, 并得到 $V_i = v_i^k$ 和 $U_i = u_i^k$ 。因此因果网中原子干预的 MPE 实例的计算可以归约为对应贝叶斯网络中 MPE 实例的计算。采用对结点 V_i 完全原子干预的结果, 可以计算对应贝叶斯网络中 V_i 的实例。

定理 2 在得到结点 V_i 完全原子干预的结果后, 对应贝叶斯网络可以采用如下方法计算 MPE 概率:

1) 如果 V_i 是非根结点, 则有如下等式:

$$MPE_p(\mathbf{e}) = \max(P(v_i | \mathbf{u}_i) MPE_p(\mathbf{e} | do(v_i))) \quad (7)$$

2) 如果 V_i 是根结点, 则有如下等式:

$$MPE_p(\mathbf{e}) = \max(P(v_i) MPE_p(\mathbf{e} | do(v_i))) \quad (8)$$

此时, 得到一个 MPE 实例 $V_i = v_i^k$ 和 $U_i = u_i^k$ 。这里, $\max(P(v_i | \mathbf{u}_i) MPE_p(\mathbf{e} | do(V_i)))$ 表示在结果中所有元组的最大概率。

证明:

1) V_i 是非根结点。在因果网中完成对非叶子结点 V_i 的 MPE 的完全原子干预, 得到等式(3), 将其两边都乘以 $P(v_i | \mathbf{u}_i)$ 可得如下等式:

$$P(v_i | \mathbf{u}_i) MPE_p(\mathbf{e} | do(V_i)) = P(v_i | \mathbf{u}_i) \max_{\lambda \setminus \{v_i, \mathbf{e}\}} \prod_{j \neq i} P(v_j | \mathbf{u}_j)$$

我们进一步推导如下:

$$\begin{aligned} MPE_p(\mathbf{e}) &= \max(P(v_i | \mathbf{u}_i) \max_{\lambda \setminus \{v_i, \mathbf{e}\}} \prod_{j \neq i} P(v_j | \mathbf{u}_j)) \\ &= \max(P(v_i | \mathbf{u}_i) MPE_p(\mathbf{e} | do(V_i))) \end{aligned}$$

因此在结果 $P(v_i | \mathbf{u}_i) MPE_p(\mathbf{e} | do(V_i))$ 中, 所有元组的最大概率为对应贝叶斯网络的 MPE 概率。类似地, 如果 V_i 是叶子结点, 非常容易得到等式 $MPE_p(\mathbf{e}) = \max(P(v_i | \mathbf{u}_i) MPE_p(\mathbf{e} | do(V_i)))$ 。由此, 通过积连接生成了 MPE 概率并且属于 $P(v_i | \mathbf{u}_i)$ 和 $MPE_p(\mathbf{e} | do(V_i))$ 的元组, 包含实例 $V_i = v_i^k$ 和 $U_i = u_i^k$ 。

2) V_i 是根结点。采用上述类似方法, 可以证明等式(8)成立, 并且等式(8)为对应贝叶斯网络中的结点找到一个 MPE 实例。

定理 2 得证。

我们可以采用一个数据特性来优化 MPE 完全原子干预的微分语义的计算, 当求解等式 MPE 完全原子干预的微分语义时, 如果一个元组的概率小于 $MPE_p(\mathbf{e})$ 或者它与当前 MPE 实例不一致, 则该元组为不可满足元组。我们采用定理 3 来从积连接操作中排除不可满足元组, 该定理来源于微分 MPE 问题^[13]。

定理 3 如果 $p_1 \cdots \cdots p_n = p$, 这里 $0 \leq p_i \leq 1 (i=1, \dots, n)$ 并且 $0 \leq p_i \leq 1$, 则有 $p_i \geq p$ 。

由于 $P(v_i | \mathbf{u}_i) \leq 1$, $MPE_p(\mathbf{e} | do(V_i)) \leq 1$ 并且 $MPE_p(\mathbf{e}) \leq 1$, 因此根据定理 3, 如果 $P(v_i | \mathbf{u}_i) \leq MPE_p(\mathbf{e})$, 则 v_i 不可能是 V_i 的实例。满足 $P(v_i | \mathbf{u}_i) < MPE_p(\mathbf{e})$ 的元组不需要参加到积连接操作中; 类似地, 满足 $MPE_p(\mathbf{e} | do(V_i)) < MPE_p(\mathbf{e})$ 的元组不需要参加到积连接操作中。例 2 是定理 2 和定理 3 的应用。

例 2 延续例 1, 采用图 3 的结果, 在给定的证据 $D=1$ 时,

根据定理 2 和定理 3 计算对应贝叶斯网络的 MPE, 其 MPE 概率为 0.3837, 一个 MPE 实例为 $A=0, B=0, C=0, D=1$ 和 $E=1$ 。

我们给出定义 2 来描述 $BMPE_p(e)$ 和 $MPE_p(e)$ 的不同。

定义 2 在因果网中, 对 MPE 进行最好原子干预后, 干预的增益定义为:

$$IGain = BMPE_p(e) - MPE_p(e)$$

在图 1 的因果网中给定证据 $D=1$, 例 1 中计算了 $BMPE_p(e)$, 例 2 中计算了 $MPE_p(e)$, 因此干预的增益为 $IGain = BMPE_p(e) - MPE_p(e) = 0.7374 - 0.3837 = 0.3537$ 。等式(7)揭示了对 MPE 完全原子干预的微分语义, 本文没有把因果网中的 MPE 计算转化为对应贝叶斯网络中对 MPE 的微分后采用式(7)计算 $MPE_p(e|do(V_i))$ 的方式。然而, 本文采用了式(3)和式(4), 并引入了联合树算法 BJT 来直接计算因果网中对 MPE 采用的最好原子干预。

下面简要地解释 BJT 算法的流程。该算法首先根据网络结构和证据 e 编辑联合树, 联合树中消息的传递分为两个阶段, 等消息传递完成后对结点进行干预。BJT 采用 $|C_i|$ 表示簇 C_i 中的结点个数; 采用式(3)、式(4)和一个缓存的因子 Ψ_i 来计算对各个簇 C_i 中所有结点的完全原子干预。因此, 该算法不需要每次都从头开始计算干预的效果, 而是从缓存的因子处开始计算。当完成对一个结点完全原子干预的计算后, 采用定理 2 和定理 3 来计算对应贝叶斯网络中的实例。该算法可以在各个簇中并行地计算完全原子干预。在对所有结点计算完全原子干预后, 计算最好原子干预 $do(V_i^k)$ 的值 $BMPE_p(e)$ 。采用定理 1 中证明的方法计算实例 $U_i = u_i^k$ 。一旦设置 $V_i = v_i^k$ 和 $U_i = u_i^k$, 则重新向上和向下传递消息。最后 BJT 算法采用定理 1, 通过计算对应贝叶斯网络 $MPE(e)$ 的实例 $V_i = v_i^k$ 和 $U_i = u_i^k$ 来计算因果网中的 BMPE 实例。BJT 利用联合树来计算因果网中的最好原子干预, 因此其时间复杂度为 $O(n * \exp(w) + n_1^2 * l * \exp(w_1))$, 这里 n 表示结点个数, w 表示联合树的宽度, l 表示联合树中簇的个数, n_1 表示最大簇中结点的个数, w_1 表示所有簇中的最大联合树宽度。

算法 1 BJT 算法

输入: CBN, e

输出: MPE 实例、BMPE 实例和 IGain

1. 编辑联合树和两阶段消息传递

$$M_{ij} = \max_{c_i \setminus s_j} \Phi_i \prod_{k \neq j} M_{k,i};$$

2. for(每一个簇 C_1);

3. 对每个簇选择一个干预序 π_i ;

4. for(每一个结点 $V_j \in \pi_i$);

5. if(V_j 是第一个结点) $\Psi_i = 1$;

6. 设 $\Phi = \prod_k M_{k,i} \prod_{l=i+1}^{|C_i|} P(v_l | u_l) \Psi_i$;

7. if(V_j 是一个非叶结点);

8. $MPE_p(e|do(V_j)) = \max_{\gamma \setminus (v_j, e)} \Phi$;

9. else $MPE_p(e|do(V_j)) = \max_{\gamma \setminus e} \Phi$;

10. 采用定理 2 和定理 3 计算 MPE 实例 $V_i = v_i^k$ 和 $U_i = u_i^k$;

11. 设置 $\Psi_i = \Psi_i P(v_i | u_i)$;

12. 计算 $BMPE_p(e)$ 和 IGain;

13. 为最好原子干预设置 $V_i = v_i^k$ 和 $U_i = u_i^k$;

14. 重新向上和向下传递消息;

15. 采用定理 1 计算 BMPE 实例。

例 3 是算法 1 的应用。

例 3 从例 1 可以知道最好的原子干预为 $do(A=1)$ 。因为 A 是根结点, BJT 仅仅设置 $A=1$, 接着重新向上和向下传递消息。然后 BJT 采用定理 1 来计算 BMPE 实例: $A=1, B=0, C=0, D=1$ 和 $E=1$ 。我们可以非常容易地区分因果网的 BMPE 实例和对应贝叶斯网络的 MPE 实例。

4 实验

在 BNJ^[14] 的基础上, 本文实现了 BJT 算法。ETree^[3] 是计算近似干预的。我们比较了 ETree 和本文提出的 BJT 计算最好原子干预的性能, 包含计算 BMPE 概率和 BMPE 实例。如果一个算法既不采用因子来缓存中间结果, 也不利用定理 3 来优化各个贝叶斯网络中 MPE 实例的计算, 则称其为 MPEI 算法。我们还比较了 BJT 和 MPEI 以便计算优化效果, BJT 和 MPEI 都是计算干预后的精确值。实验中采用的因果网是来自相关标准^[15]。本文把执行时间分为编辑时间和推理时间来揭示 MPEI 和 BJT 的特性, 并在表 1 中采用 compile 来表示 MPEI 和 BJT 在编辑网络到联合树所花费的时间, 分别采用 MPEI-I 和 BJT-I 来表示 MPEI 和 BJT 在推理阶段所花费的时间。

表 1 ETree 和 BJT 以及 MPEI 的性能对比

Table 1 Performance comparison of ETree, BJT and MPEI

(单位: ms)

Networks	Compile	BJT-I	MPEI-I	ETree
alarm	8.1	4.8	7.4	84.2
bm_05_01	521	43.6	96.4	72940
bm_05_02	812	58.4	118.8	138852
bm_05_03	893	57.2	126.4	179493
diabetes	42	40.3	94.2	19664
fs-04	19.8	4.5	7.2	1037
hailfinder	11	5.6	8.8	245.3
mm_3_8_3	731	32.2	70.4	178364
mumin1	638	78.4	160.8	14193
mumin2	74	45.4	86.8	39782
mumin3	97	61.6	131.2	54349
mumin4	113	73.2	128.4	63811
pathfinder	40	19.4	42.8	1792
pigs	36	14.8	31.8	4762
students-3_2	74	18.4	43.8	5564
water	22.2	10.8	16.2	257.5

从表 1 观察到, MPEI 和 BJT 在绝大多数因果网上执行的性能都比 ETree 好, 原因是 ETree 在执行最好近似原子干预前要生成 $\sum_{i=1}^n |V_i|$ 个解释树, 而 MPEI 和 BJT 只需要编辑一个联合树来计算最好原子干预。如果对每个贝叶斯网络有一个证据, 从表 1 可以观察到, 在计算对所有结点的完全原子干预和 BMPE 实例方面, BJT 在各个因果网中的效果都比 MPEI 更好。

因为 MPEI 和 BJT 都是基于联合树算法的, 它们的编辑

阶段是一样的。在推理阶段,我们发现 BJT-I 比 MPE-I 的性能要好很多,因此优化效果更为明显。这是因为 BJT-I 有以下两个特性:1)BJT-I 的干预不需要每次都从零开始计算,而是从缓存的因子处开始计算;2)BJT-I 可以充分利用数据特性来优化贝叶斯网络中 MPE 实例的计算。在许多情况下,BJT-I 不需要执行它们的积连接操作就可以计算 MPE 实例。同时,从表 1 中能够观察到这两个算法都非常高效,这是因为它们具有如下 3 个特性:1)它们都可以在各个簇内并行地计算对各个结点的完全原子干预;2)这两个算法都只采用对每个叶结点的一个原子干预来计算完全原子干预;3)对于每个结点,这两个算法如果已经找到该结点的实例,那么它们不再需要计算式(7)或者式(8)右侧的式子。

结束语 因果网被广泛地用于进行因果推导,干预是因果网的核心操作,本文主要研究在因果网中对 MPE 进行干预。在揭示对 MPE 完全原子干预的微分语义后,本文提出了一种联合树算法来计算在因果网中对 MPE 的最好原子干预和对应贝叶斯网络中的 MPE 实例。为了对干预进行优化计算,我们采用一个因子来缓存中间结果,这样就不需要每次都从零开始计算。本文研究有广泛的应用场景,其中一个应用是政府可以计算对经济的最好原子干预,并以此为依据制定最适合的产业政策来刺激经济。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,并对干预结果进行合理解释。

参 考 文 献

- [1] PEARL J. Causality: models, reasoning, and inference[M]. Cambridge University Press, 2009.
- [2] LI C, QIN B. New algorithms for computing max-product instantiations[J]. Application Research of Computers, 2015, 32(6): 1711-1715.
- [3] NIELSEN U, PELLET J, ELISSEE A. Explanation trees for causal Bayesian networks[C]//Proc of the 24th Conference on Uncertainty Artificial Intelligence. 2008: 427-434.
- [4] QIN B. Differential semantics of intervention in Bayesian networks[C]//Proc of the 24th International Joint Conference on Artificial Intelligence. 2015: 710-716.
- [5] PARK J, DARWICHE A. A differential semantics for jointree algorithms [J]. Artificial Intelligence, 2004, 156(1): 197-216.
- [6] 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.
- [7] MARINESCU R, LEE J, DECHTER R. AND/OR search for marginal MAP [J]. Journal of Artificial Intelligence Research, 2018, 63: 875-921.
- [8] LOU Q, DECHTER R, IHLER A. Anytime anytime AND/OR best-first search for bounding marginal MAP[C]//Proc of the 32nd Conference on Artificial Intelligence. 2018: 6392-6400.
- [9] SHENOY P, SHAFER G. Propagating belief functions with local computations [J]. IEEE Expert, 1986, 1(3): 43-52.
- [10] 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.
- [11] DARWICHE A. Modeling and reasoning with bayesian networks [M]. Cambridge University Press, 2009.
- [12] HALPERN J. Axiomatizing causal reasoning [J]. Journal of Artificial Intelligence Research, 2000, 12: 317-337.
- [13] LI C. Jointree algorithm for the differentiation of the MPE problem [J]. Journal of Chinese Computer Systems, 2016, 37(10): 2306-2311.
- [14] <http://sourceforge.net/projects/bnj/>.
- [15] <http://reasoning.cs.ucla.edu/ace>.



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, associate professor, Ph.D supervisor, is a member of China Computer Federation. His main research interests include causal Bayesian network and machine learning.