文章编号:2095-6134(2020)01-0136-08

简 报

和积网络的性质分析及其有效性验证算法*

刘洋[1†],罗晨希[2],罗铁坚[1]

(1 中国科学院大学计算机与控制学院,北京 101408; 2 中国科学院软件所,北京 100080)

(2018 年 4 月 23 收稿; 2018 年 7 月 18 日收修改稿)

Liu Y, Luo C X, Luo T J. Property analysis and validity verification algorithms of sum-product network[J]. Journal of University of Chinese Academy of Sciences, 2019,37(1):136-143.

摘 要 和积网络(sum-product networks, SPN)是一种在多层网络中进行快速推理的深度概率图模型,在人工智能领域有广泛应用前景。SPN 的有效性即它可用来正确表示概率分布,使得 SPN 可以表示一些图模型的配分函数和所有的边缘分布。由于只有部分 SPN 是有效的,快速判断 SPN 的有效性很有必要。针对 SPN 理论体系中的有效性验证问题,讨论并揭示 SPN 内部结构性质,提出验证 SPN 有效性的两个算法,并给出算法的正确性证明及其复杂度。还通过给出一种新的 SPN 中生成树个数的计算方法来验证 SPN 有效性算法的可靠性。

关键词 深度学习;概率图模型;和积网络;有效性

中图分类号:TP305 **文献标志码**:A **doi**:10.7523/j.issn.2095-6134.2020.01.016

Property analysis and validity verification algorithms of sum-product network

LIU Yang[1], LUO Chenxi[2], LUO Tiejian[1]

(1 *School of Computer and Control, University of Chinese Academy of Sciences, Beijing* 101408, *China*;
2 *Institute of Software, Chinese Academy of Sciences, Beijing* 100080, *China*)

Abstract Sum-product networks (SPN) is a deep probabilistic graphical model which has the characteristic of fast inference in multilayer networks, and it has wide application prospect in the field of artificial intelligence. The validity of SPN is that it can be used to represent the probability distribution correctly so that SPN can be used to represent the distribution functions of some graph models and all the marginal distributions. Since SPN is not always valid, it is necessary to determine the effectiveness of SPN quickly. In this paper, we consider the problem of validity verification in the SPN theoretical system, reveal the internal structure properties of SPN, and propose two algorithms for verifying the validity of SPN. The correctness proofs and the complexity of the proposed algorithms are given. We also verify the reliability of the proposed algorithms by giving a new method of calculating the number of generation trees in SPN.

Keywords deep learning; probabilistic graphical models; sum-product networks; validity

* 中国科学院仪器共享设备管理系统(Y42901VED2)资助

† 通信作者,E-mail:liuyang810@mails.ucas.ac.cn

概率图模型,例如贝叶斯网络和马尔科夫网络,已经被广泛应用在人工智能领域[1]。这类图模型用简洁的结构表示复杂的概率分布,在推理和参数学习方面,由于归一化参数的计算量非常大,在最坏情况下为指数级别[2],一般情况下精确计算条件概率的复杂度也为#P 完全[3]的。此外,随着变量个数的增加,模型也更加复杂。Poon 和 Domingos[1]于 2011 年提出一种新型深度模型结构——和积网络(sum-product network, SPN)来解决上述问题。和积网络(图 1)是有向无环图,它将观测变量作为叶子结点,将"和"与"积"操作作为深度网络的内部结点。和积网络可以在高树宽模型中快速计算精确推理,其推理开销和网络大小成线性关系,具有很强的表达能力和快速推理能力,在计算机视觉[4]、语音识别[5]、自然语言处理[6]等领域均有应用。当前和积网络的研究主要聚焦在结构学习和参数学习等应用方面,而制约其应用发展的理论问题,如 SPN 有效性、MAP 推理复杂性等问题仍未得到根本解决。

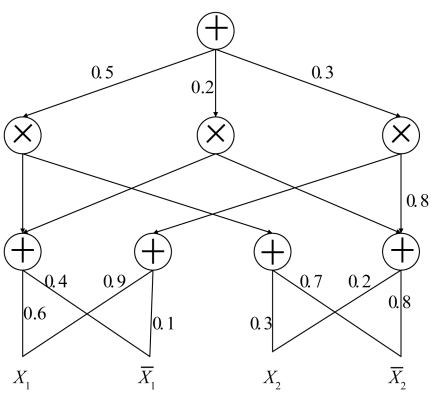


图 1　和积网络示例

Fig. 1　An example of sum-product networks

和积网络将分布分解为混合(sum)与分解(product)的层次结构,因而可看做基于变量集合的非归一化概率分布。和积网络的有效性(validity)即它可以正确表示概率分布,例如,仅当和积网络有效时,计算联合概率分布 p_x 的等式才成立:对 $\forall x, p(X=x)=S(x)/\sum_{x\sim X}S(x)$,$X$ 表示变量集合,$S(x)$ 表示输入为 x 的和积网络计算得到的数值。在实践中并非所有生成的和积网络都是有效的,和积网络结构学习和参数学习过程中,部分算法基于有效性对于范围的限制,应用分层联合聚类[7-9],每迭代一次,都需对现有模型进行有效性验证,因而快速判断和积网络的有效性成为必要,开展和积网络有效性问题的研究具有理论意义和应用价值。

本文的主要工作及其贡献在于:1)探讨和积网络的一些内部结构性质;2)提出验证和积网络的有效性算法及其正确性证明,分析其复杂度和适用场景;3)提出一种新的关于和积网络中生成树个数的计算方式。

1　相关工作

和积网络被提出后在基础理论方面、参数学习、结构学习方面以及应用方面都有了相应的研究,从和积网络的基础理论研究方面,Zhao 等[10]证明任意和积网络都可以转化为二分贝叶斯网络。Martens 和 Medabalimi[11]证明网络的表达性随着深度增长而增加。和积网络有效性的定义给出后[1],由于没有相应的验证算法,Peharz[2]弱化有效性的条件要求,证明和积网络的参数可以被转化为局部归一化。一致且完备的和积网络同样保证有效性[12]。Martens 和 Medabalimi[11]找到有效和积网络的特例,证明 Peharz 的结论是充分不必要条件,因而强化了有效性的条件,给出强有效性的定义,并证明强有效性与网络输出多重线性多项式的等价关系,而关于和积网络有效性的算法验证的研究仍不充分。本文尝试给出两个验证算法,并比较两个算法的适用场所及复杂度。

从和积网络的权重学习方面,Zhao 等[13]提出一种统一的学习和积网络参数的方法。

从和积网络的结构学习方面,Dennis 和 Ventura[14]提出和积网络的第一个结构学习算法,Adel 等[15]提出一种基于 SVD 分解的算法,通过对样本矩阵进行切割学习和积网络的结构。Rashwan 等[16]提出在线贝叶斯矩匹配算法。目前最为常用的算法为 LearnSPN[7],虽然提出较早,但由于执行速度快,至今仍被频繁使用,也出现对其改进的算法 SPN-BT[8]。目前最好的结构学习算法为 ID-SPN[9],通过利用变量之间的间接和直接相互作用、自上而下聚类的方式学习和积网络,该算法将 sum 结点和 product 结点作为内部结点,运算电路 AC(arithmetic circuits)作为叶结点,从单个 AC 结构开始,只在直接改善了似然性的情况下将每个 AC 叶结点拆分为两个新的 AC 叶结点,最后估计叶结点的分布。由于该算法与 AC 的基础算法相结合,因而比一般的树形结构如 Chow-Liu 叶结点更好地逼近复杂的分布。

从应用方面,Cheng 等[6]利用和积网络构造语言模型,根据前 K 个词计算第 $K+1$ 个词出现的概率,得到比之前语言模型更好的预测准确性。Nath 和 Domingos[17]引入关系和积网络的概念,并利用关系和积网络构造一个关于缺陷定位问题的易解概率学习模型,可以鉴别重复发生的 bug。

和积网络可以进行概率密度估计,因而也被解释为特殊的深度神经网络。同时,它还可以表达一些其他类型的易解概率分布,如稀疏联结树和隐藏树模型。近期,和积网络也被解释为特殊形式的前馈神经网络[4]和概率卷积网络[18]。

和积网络经常被称作一种新型的概率图模型,其性质更像是运算电路。在经典的概率图模型中,叶结点代表随机变量,边代表变量间的依赖关系。然而和积网络的中间结点为运算单元,即“和”与“积”,边决定运算单元的计算顺序,不再关心变量间的关系。

2 和积网络的有效性验证算法

和积网络是有向无环图(如图 1),隐藏变量为求和或者求积,并且被交替排列在相邻层次上,sum 结点可看作是变量在集合上的混合,product 结点可看作是特征的混合,即 sum 结点提供混合模型,product 结点建立特征层次,“有效性”以一种很好的方式约束和积网络,因而网络具有很强的表达能力和推理能力。每一个以和积网络内部结点为根结点的子网络依然为和积网络,和积网络可看作由小的和积网络连接组成,因而在计算上具有潜在的可扩展性,使得学习和推理更加便于处理,在该模型下得到的结果也应用得更加广泛。

现在给出和积网络的形式化定义。

定义 1 和积网络[1]:和积网络是一个有根且有权重的有向无环图,内部结点为 sum 结点与 product 结点,叶结点为网络输入,输入变量集合为 $X=\{X_1,\cdots,X_n\}$,X 的分布为 ϕ_n,$|X|=n$。

令 $\mathrm{ch}(Q)$ 作为结点 Q 的所有子结点的集合,$\mathrm{pa}(Q)$ 作为结点 Q 的所有父结点的集合,$\mathrm{desc}(Q)$ 为结点 Q 的所有后代的集合。每条连接 sum 结点 Q 与其子结点 $c\in\mathrm{ch}(Q)$ 的边均有非负权重 ω_{Qc},令 w 为和积网络所有权重的集合。S_Q 是 S 的子网络,其根结点为 Q。网络内所有结点个数设为 m。对于 X 的一个实例 x,或者称之为完全示例(complete evidence),用 $x\sim X$ 表示。为简化讨论,本文假定随机变量为布尔变量,然而本文结论可被拓展到离散和连续变量。不失一般性,本文假设和积网络有交替的内部结点类型,即 sum 结点层与 product 结点层交替出现。

叶结点的范围(scope)[1]是结点对应向量的集合,父结点的范围是所有子结点范围的并集,并把根结点的范围作为和积网络的范围。

和积网络的深度是为 sum 结点层与 product 结点层交替出现的和积网络中从根结点到子结点的最长路径长度[1]。

当变量集合 X 取值为 x 且和积网络(以下用 S 表示)有效时,$S(x)$ 表示输入为 x 基于 S 计算得到的非归一化概率值,即 $P(X=x)=S(x)/Z$,$Z=\sum_x S(x)$ 为归一化参数。当每个 sum 结点 Q_i 连接所有子结点的权重和为 1 时,即 $\sum_{N_j\in\mathrm{ch}(N_i)}\omega_{Q_iN_j}=1$,且叶结点的分布为归一化分布,则 S 表示归一化分布,即 $\forall x, P(X=x)=S(x)$。

$S(x)$ 在网络中的计算方式为自下而上,x 作为叶结点的输入,sum 结点的值为子结点值的加权和,product 结点的值为子结点值的乘积,最终得到根结点的值作为 $S(x)$ 的值。如图 1,当 $X_1=1, X_2=0, X_1=0, \bar{X}_2=1$ 时,则根结点的值为 $0.5\times(0.6\times1+0.4\times0)(0.3\times0+0.7\times1)+0.2\times(0.6\times1+0.4\times0)(0.2\times0+0.8\times1)+0.3\times(0.9\times1+0.1\times0)(0.2\times0+0.8\times1)=0.522$,并把根结点的值作为整个网络的值。下面给出和积网络中计算的形式化定义。

定义 2 计算(evaluation)[1]:对于和积网络 S,给定变量集合 X,X 的分布为 ϕ_n,$|X|=n$,参数集合 w,若 S_Q 是 S 的子网络,其根结点为 Q,则 $S_Q(x)$ 的计算公式为

$$S_Q(x)=\begin{cases}\phi_n(\mathrm{sc}(Q)=x_{|\mathrm{sc}(Q)}), & \text{若 } Q \text{ 为叶结点},\\ \sum_{c\in\mathrm{ch}(Q)}\omega_{Qc}S_c(x), & \text{若 } Q \text{ 为 sum 结点},\\ \prod_{c\in\mathrm{ch}(Q)}S_c(x), & \text{若 } Q \text{ 为 product 结点}.\end{cases}$$

整个网络的值 $S(x)$ 为根结点的值。

定义 3 网络多项式(network polynomial)[13]:令 $f(\cdot)\geqslant0$ 为基于布尔变量集合 X 上的概率分布,$f(\cdot)$ 的网络多项式是一个关于指示变量的多线性方程 $\sum_x f(x)\prod_1^n I_x$,其中指示变量 I_x 当 $X=x$ 时为 1,否则为 0。

和积网络的有效性即 S 可以正确表示分布,下面给出有效性的形式化定义。

定义 4 有效性[10]:对于和积网络 S,给定变量

集合 X,若对于任意 $x \sim X$,均有 $S(x)=\phi_S(x)$,则称 S 是有效的,其中,$\phi_S(x)$ 为基于 S 的网络多项式。

定义 5　完备性,可分解性[1]:和积网络 S,令 S_+ 为 S 中所有 sum 结点的集合,$S_\times$ 为 S 中所有 product 结点的集合,则

1) S 是完备的当且仅当 $\forall n \in S_+, \forall c_1, c_2 \in \mathrm{ch}(n): \mathrm{sc}(c_1)=\mathrm{sc}(c_2)$;

2) S 是可分解的当且仅当 $\forall n \in S_\times, \forall c_1, c_2 \in \mathrm{ch}(n), c_1 \neq c_2$:

$$\mathrm{sc}(c_1) \cap \mathrm{sc}(c_2)=\varnothing。$$

定义 6　拓扑排序(topological sorting)[13]:在图论中,由一个有向无环图的顶点组成的序列,当且仅当满足下列条件时,称为该图的一个拓扑排序。

1)每个顶点出现且只出现 1 次;

2)若 A 在序列中排在 B 的前面,则在图中不存在从 B 到 A 的路径。

2.1　Veri-SPN 算法

算法 1 对每个结点都进行验证,计算量较大。首先自下而上标记所有结点的范围,再自上而下,逐层分别检查每个结点的完备性和可分解性,即对于 sum 结点,检查其子结点的范围是否相同,对于 product 结点,检查其子结点的范围是否互不相交,通过验证和积网络的完备性和可分解性进而验证有效性,第 1 个算法为 Veri-SPN,具体描述如下。

定理 2.1　对结点个数为 m,变量个数为 n 的和积网络 S,算法一在 $\mathrm{O}(n \times m^2)$ 内完成验证。

证明　算法 1 的证明方式为自下而上,设 $Q_1, \cdots, Q_m$ 是 S 所有结点的一个拓扑排序,即 $k > l \Rightarrow Q_k \notin \mathrm{desc}(Q_l)$。首先对每个结点 Q_k 初始化赋值为 0:$\boldsymbol{a}_k=[a_{k1}, a_{k2}, \cdots, a_{kn}]=[0,0,\cdots,0]$,该向量的含义是:若 $X_i \in \mathrm{sc}(Q_k)$,则 $a_{ki}=1$,否则为 0。即,若 $X_i \in \mathrm{sc}(Q_k)$,则 $\boldsymbol{a}_k$ 在第 i 个位置的值为 1。

当结点 Q_k 为叶结点时,需将该叶结点对应的向量反映在自己的矩阵中,即,若 $X_1 \notin \mathrm{sc}(Q_k)$,$X_2 \in \mathrm{sc}(Q_k) \Rightarrow \boldsymbol{a}_k=[0,1,\cdots,0]$,由于本文假设变量类型为布尔变量,变量个数为 n,结点个数为 m,由于叶结点个数可能超过 n,并且与 n 并无明显关系,但是却一定小于 m,则所有叶结点的计算成本为 $O(m)$。

当结点 Q_k 为 product 结点时,需要验证 Q_k 的子结点范围互不相同,即 $\forall Q_i, Q_j \in \mathrm{ch}(Q_k), i \neq j, \mathrm{sc}(Q_i) \cap \mathrm{sc}(Q_j)=\varnothing$,首先逐个找到 Q_k 的子结点,然后将子结点的范围并入到 Q_k 的范围,若并入过程中发现已经存在重复变量,则 S 不满足可分解性,进而不是有效的,则算法停止,所有的 product 结点的计算成本为 $O(n \times m^2)$。

```
Algorithm 1　Veri-SPN
1: Find some topological ordering Q_1,…,Q_m of nodes in SPN
2: For all nodes Q_k initialize a_k=[a_k1,a_k2,…,a_kn]=[0,0,…,0],a_ki
∈[0,1]
3: for k = 1:m do
4:    b←0
5:    if Q_k is a leaf node then
6:       if X_i ∈ sc(Q_k), i = 1:n then
7:          a_ki←1
8:       end if
9:    end if
10:   if Q_k is a product node then
11:      if Q_i ∈ ch(Q_k), i = 1:k then
12:         if a_ij = 1, j = 1:n then
13:            if a_kj = 1 then
14:               abort
15:            else
16:               a_kj←a_ij
17:            end if
18:         end if
19:      end if
20:   end if
21:   if Q_k is a sum node then
22:      if Q_i ∈ ch(Q_k), i = 1:k then
23:         if b = 0then
24:            a_kj←a_ij, j=1:n, b←1
25:         else
26:            if a_kj ≠ a_ij, j = 1:n then
27:               abort
28:            end if
29:         end if
30:      end if
31:   end if
32: end for
```

当结点 Q_k 为 sum 结点时,需要验证 Q_k 的子结点范围完全相同,即 $\forall Q_i, Q_j \in \mathrm{ch}(Q_k), \mathrm{sc}(Q_i)=\mathrm{sc}(Q_j)$。首先把第一个得到的子结点的范围并入到 Q_k 的范围,之后逐个比较 Q_k 的每个子结点的范围与 Q_k 的范围,若出现不同,则 S 不满足完备性,进而不是有效的,则算法停止,所有的 sum 结点的计算成本为 $O(n \times m^2)$。

若和积网络是完备的且是可分解的,则该和积网络是有效的[1],因而算法 1 可以证明和积网络的有效性。

综上,总的计算成本为 $O(n \times m^2)$。　□

由于目前没有任何关于 m 与 n 关系的探讨,m 的取值范围可以是 n 的多项式范围,也可以是 n 的指数范围。如果是前者,则算法 1 在实践中已满足使用需求;相反,如果是后者,在 m 相当大的情况下,算法 1 计算量过大。

因而,本文提出第 2 个验证算法,牺牲一部分

准确性,换取相对快速的验证效果,在 S 的层数低时效果更好。

2.2 Induce-SPN 算法

有效的和积网络可看做是一些生成树 T (induced tree)的叠加,且每个生成树的范围与对应和积网络是相同的,因而对和积网络的有效性验证就可以一定概率转化为对 T 进行验证,算法 2 基于上述思想,具体描述如下。

Algorithm 2 Induce-SPN

1: Let $T = (T_V, T_E)$ be a subgraph of S
2: $T_V = \{\text{Root(S)}\}$, $T_E = \varnothing$
3: while exist node $Q_k \in T_V$ that $ch(Q_k) \neq \varnothing$, $\text{ch}(Q_k) \cap T_E = \varnothing$ do
4: if Q_k is a sum node then
5: choose exactly one child of Q_k in S randomly, put the node in T_V, the responding edges in T_E
6: end if
7: if Q_k is a product node then
8: put all the children of Q_k in S in T_V in order, the responding edges in T_E
9: end if
10: if Q_k is a leaf node then
11: skip
12: end if
13: $k \leftarrow k+1$
14: end while

定理 2.2 S 的范围与 T 的范围是一样的,并且, T 中每个结点的范围与对应的 S 中结点的范围是一样的。

证明 由于 S 的根结点 Root(S) 也是 T 中的结点,

1) 当 S 的高度 $h = 2$ 时,若 Root(S) 是 sum 结点,则根结点与相邻子结点的范围是一样的,显然, $\text{sc}(S) = \text{sc}(T)$, 若 Root($S$) 是 product 结点,则 Root($S$) 的所有相邻子结点也在 T 中, $S = T$, 因而 $\text{sc}(S) = \text{sc}(T)$。

2) 假设当 $h = n$ 时, $\text{sc}(S) = \text{sc}(T)$, 那么,当 $h = n + 1$ 时,若 Root(S) 是 sum 结点,有 k 个子结点,则 S 与每个子网络的根结点的范围一样,同理于 T, 因而 $\text{sc}(S) = \text{sc}(T)$。 若 Root($S$) 是 product 结点,则 Root($S$) 与子结点连接的边也在 T 中,因而 $\text{sc}(S) = \text{sc}(T)$。

综上, $\text{sc}(S) = \text{sc}(T)$。

用同样的方法可以得到, T 中每个结点的范围与对应的 S 中结点的范围是一样的。 □

定理 2.3 若和积网络 S 是不可分解的,则存在 T 也是不可分解的。

证明 首先,若 S 是有效的,则 T 也是有效的。由于 T 中所有的 sum 结点均只有一个子结点,完备性要求 sum 结点的所有子结点的范围完全相同,因而 T 天然地满足完备性。T 中任意 product 结点与其子结点均来自于 S 中的一个 product 结点及其所有的子结点,由于 S 是有效的,则所有子结点的范围是互不相交的,由于 T 的生成过程, T 是 S 的子图, T 中每个结点在 T 中的范围均小于等于在 S 中的范围,因而,在 T 中所有子结点的范围依旧是不相交的,满足可分解性要求,所以, T 也是有效的。

若 S 是不可分解的,则存在 product 结点 Q_k, 其子结点的范围是有交集的,由定理 2.2, T 中每个结点的范围与对应的 S 中结点的范围是一样的。因而,若 T 包含结点 Q_k, 则 T 包含 Q_k 所有子结点,因而 T 中 Q_k 的子结点的范围是有交集的, T 不可分解。 □

定理 2.4 若 S 不是完备的,则存在 T 是不可分解的。

证明 若 S 不是完备的,则存在结点 Q_i、Q_j 有共同的父结点 Q, Q 为 sum 结点且 $\text{sc}(Q_i) \neq \text{sc}(Q_j)$。 由于 T 中每个 sum 结点只连接一条边,因而 Q_i、Q_j 不会出现在同一个 T 中,则存在 T_i、T_j 分别包含边 (Q, Q_j) 与边 (Q, Q_i)。

由于 $\text{sc}(Q_i) \neq \text{sc}(Q_j)$, 父结点的范围是子结点范围的并集,因而 Q 在 T_i 中与 Q 在 T_j 中的范围不一样,然而根据定理 2.2, T_i 与 T_j 的范围是一样的(均与 S 相同),则根据生成树的生成规则, T_i 与 T_j 至少有一个是不可分解的。 □

定理 2.5 若和积网络 S 不是有效的,则算法 2 至少以 $1/f_S(\mathbf{1} \mid \mathbf{1})$ 的概率证明 S 的无效。

证明 尽管存在极端情况,即和积网络 S 是有效的,但是却不满足完备性或可分解性[11],由于数量极少,故在一般使用中忽略此情况。综上,若 S 是不可分解的,则存在 T 是不可分解的,同样,若 S 是不完备的,也存在 T 是不可分解的,由于 S 中所有不同的 T 的个数为 $f_S(\mathbf{1} \mid \mathbf{1})$[13], 根据定理 2.2、2.3、2.4,算法 2 至少以 $1/f_S(\mathbf{1} \mid \mathbf{1})$ 的概率证明 S 的无效。 □

算法 2 的计算复杂度为 $O(n^{\left[\frac{h}{2}\right]})$:T 中每个 sum 结点只连接一条边,因而 sum 结点连接边数总和等于 T 中 sum 结点数目,由于 product 结点的子结点均为 sum 结点,因而 T 中 sum 结点数目等于 product 结点连接边数总和。对于每个 product

结点,其子结点的范围互不相交,因而其子结点数小于等于 n, 由于 sum 结点层与 product 结点层交替出现,因而 product 结点连接边数总和小于等于 $n + n^2 + n^3 + \cdots + n^{[h/2]}$,$h$ 为 S 的高度,因而算法 2 的计算复杂度为 $O(n^{[h/2]})$。 高度比较低的和积网络中算法 2 要更好些。

下面给出两个算法的具体比较(如表 1 所示)。

表 1　两个算法比较

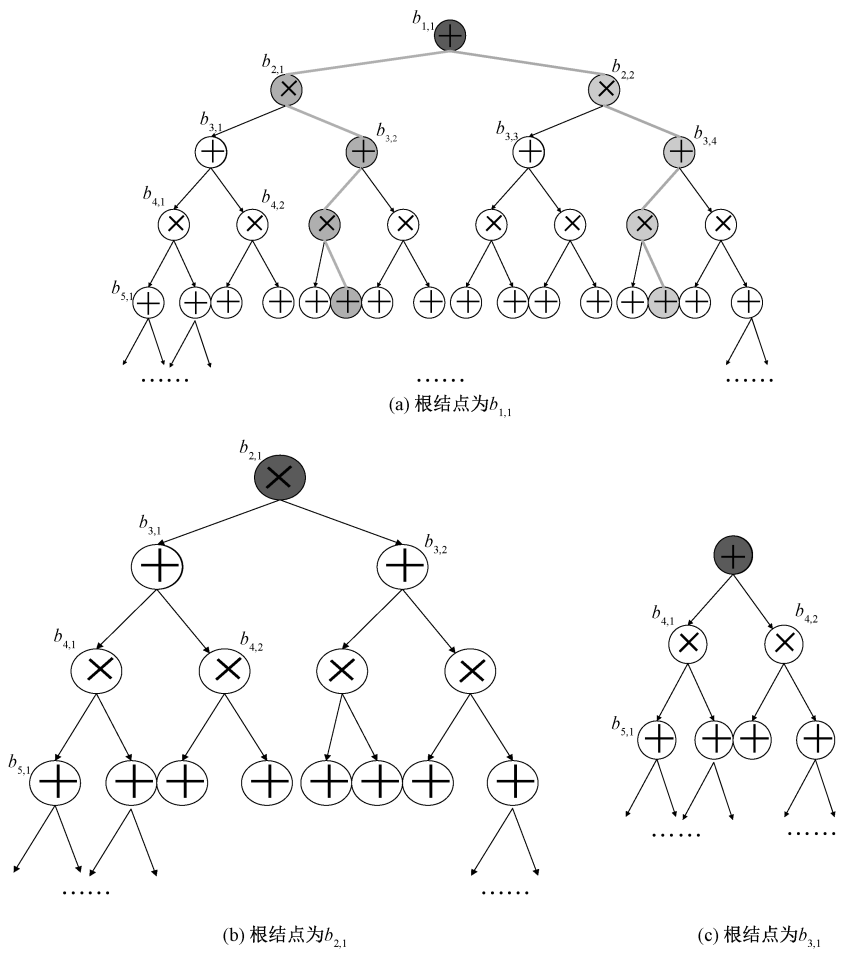
Table 1　Comparison between the two algorithms

	Veri-SPN	Induce-SPN
计算复杂度	$O(n \times m^2)$	$O(n^{[h/2]})$
优点	有效性的判断置信度高	有效性判断的置信度以概率形式表达
计算量	较大	较小
适用场景	结点个数少	网络高度小

3　一种新的计算和积网络生成树个数的方法

和积网络中的生成树 T 为算法 2 中产生的树:自上而下,首先包含 S 的根 Root(S),若为 sum 结点则 T 只包含其中的一个子结点,若为 product 结点则其所有子结点均在 T 中,以此类推,直至有一个叶结点包含在 T 中。Zhao 等[13]提到 S 中 T 的个数 τ_S 仅依赖网络结构,并且远小于 $2^{O(M)}$, 其中 M 为 S 中 sum 结点的个数,并给出 $\tau_S = \Omega(2^h)$,h 为 S 的高度,然而,本文找到的高度为 h 的最简单结构的和积网络的 T 个数为 $2^{2^{h/2}}$, 即:任意高度为 h 的和积网络,可以得到的个数大于等于 $2^{2^{h/2}}$, 因而得出更加准确的结论:$\tau_S = \Omega(2^{2^{h/2}})$。

计算过程如下:不失一般性,设 Root(S) 为 sum 结点(product 结点情况类似,可以类推),设 S 中 sum 结点层与 product 结点层交替出现,并且每个结点的子结点数为 2(以此达到最简单的构造 S),自上而下对每个结点 N_{ij} 赋予变量 $b_{i,j}$,i 代表层数($i \leqslant h$), j 代表自左向右的顺序,如图 2 (a), $b_{i,j}$ 的值代表以结点 N_{ij} 为根结点的子网络 S_{ij} 中 T 的个数。



(a) 根结点为$b_{1,1}$

(b) 根结点为$b_{2,1}$

(c) 根结点为$b_{3,1}$

图 2　和积网络生成树个数计算步骤

Fig. 2　Calculation steps for the number of generation trees in SPN

由于 S 是对称结构,任意包含边(N_{11},N_{21})的 T(深灰色)都可以在包含边(N_{11},N_{22})的 T^*(浅灰色)找到一条来对应,且由于 T 的构造,不存在 T 同时包含边(N_{11},N_{21}),(N_{11},N_{22})。因而 $b_{2,1}=b_{2,2}$,$b_{1,1}=2\cdot b_{2,1}$。如图 2(b),以 N_{11} 为根结点的计数问题转化为以 N_{21} 为根结点的子网络 S_{21} 中 T 的计数问题。

由于 N_{21} 是 product 结点,因而(N_{21},N_{31}),(N_{21},N_{32})均包含在 S_{21} 中的任意 T 中,因而 $b_{2,1}=b_{3,1}\cdot b_{3,2}$,由于 S_{21} 的对称结构,可得到 $b_{3,1}=b_{3,2}$,$b_{2,1}=b_{3,1}^2$。因而,如图 2(c),以 N_{21} 为根结点的计数问题转化为以 N_{31} 为根结点的子网络 S_{31} 中 T 的计数问题。

由于 N_{31} 是 sum 结点,再次得到 $b_{3,1}=2\cdot b_{4,1}$。

综上,得到 $b_{2k+1,1}=2\cdot b_{2k+2,1}$,$b_{2k,1}=b_{2k+1,1}^2$。由于叶结点中 T 的个数只有一个,因而 $b_{h,1}=1$。

得到
$$\begin{aligned}\tau_S=b_{1,1}&=2\cdot b_{2,1}=2\cdot b_{3,1}^2\\&=2\cdot(2b_{4,1})^2\\&=2\cdot 2^2 b_{5,1}^{22}\\&=2^{2^0}\cdot 2^{2^1}\cdot 2^{2^2}\cdot b_{6,1}^{2\cdot 2}\\&=2^{2^0}\cdot 2^{2^1}\cdot 2^{2^2}\cdot b_{7,1}^{2^3}\\&=\cdots.\end{aligned}$$

若 h 为奇数,则
$$\begin{aligned}\tau_S&=b_{1,1}\\&=2^{2^0}\cdot 2^{2^1}\cdot 2^{2^2}\cdot\cdots 2^{2^{(h-3)/2}} b_{h,1}^{2^{(h-1)/2}}\\&=2^{2^0+2^1+2^2+\cdots+2^{(h-3)/2}}=2^{2^{(h-1)/2}-1},\end{aligned}$$

若 h 为偶数,则
$$\begin{aligned}\tau_S&=b_{1,1}\\&=2^{2^0}\cdot 2^{2^1}\cdot 2^{2^2}\cdot\cdots 2^{2^{h/2}} b_{h,1}^{2^{h/2}}\\&=2^{2^0+2^1+2^2+\cdots+2^{h/2}}=2^{2^{(h+2)/2}-1}.\end{aligned}$$

综上,$\tau_S=\Omega(2^{2^{h/2}})$。

4 总结

和积网络可以简洁地表示各种边缘分布,其模型的表达能力强,推理速度快,具有广泛应用前景。针对和积网络理论体系中的有效性验证问题,通过分析和积网络的内部结构性质,给出验证和积网络有效性的两个算法及其适用场景,并通过定理证明确保算法的正确性。还提出一种新的和积网络中生成树个数的计算方法。

论文算法 1 的计算复杂度为 $O(n\times m^2)$,与网络结点个数和变量个数相关,利用和积网络能够快速计算边缘分布的特点进行构造。算法 2 的计算复杂度为 $O(n^{[h/2]})$,与网络高度和变量个数相关,首先分析和积网络的内部结构性质,找到网络与生成树之间的关系,根据结构特点进行构造。

现已发现和积网络是特殊形式的前馈神经网络和概率卷积网络。和积网络的应用已经不限制在概率图模型的框架中,将逐渐与深度神经网络一样有广泛的影响。但和积网络的理论体系仍需完善,例如,和积网络中的 MAP 推理的复杂性,学习和积网络的过程中合适的消耗函数是否存在等问题。

参考文献

[1] Poon H, Domingos P. Sum-product networks: a new deep architecture[C]//Proceedings of 12th Conf on Uncertainty in Artificial Intelligence, Barcelona, Spain: AUAI, 2011:2551-2558.

[2] Peharz R. Foundations of sum-product networks for probabilistic modeling[D]. Graze: Medical University of Graz, 2015.

[3] Roth D. On the hardness of approximate reasoning[J]. Artificial Intelligence, 1996, 82:273-302.

[4] Peharz R, Geiger B, Pernkopf F. Greedy part-wise learning of sum-product networks[C]// Machine Learning and Knowledge Discovery in Databases, Berlin, German: Springer, 2013, 8189:612-627.

[5] Peharz R, Kapeller G, Mowlaee P, et al. Modeling speech with sum-product networks: application to bandwidth extension[C]//International Conference on Acoustics, Speech and Signal Processing, Piscataway, NJ: IEEE, 2014:3699-3703.

[6] Cheng W C, Kok S, Pham H V, et al. Language modeling with sum-product networks[C]// Interspeech, Singapore, 2014: 2098-2102.

[7] Gens R, Domingos P. Learning the structure of sum-product networks[C]//Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA:ACM, 2013:873-880.

[8] Vergari A, Mauro N D, Esposito F. Simplifying, regularizing and strengthening sum-product network structure learning[C]// Proceedings of Machine Learning and Knowledge Discovery in Databases, Berlin, German: Springer, 2015: 343-358.

[9] Rooshenas A, Lowd D. Learning sum-product networks with direct and indirect variable interactions[C]//International Conference on Machine Learning, Atlanta, GA, USA:ACM, 2014, 32: 710-718.

[10] Zhao H, Melibari M, Poupart P. On the Relationship between Sum-Product Networks and Bayesian Networks[C]// Proceedings of International Conference on Machine Learning, Atlanta, GA, USA:ACM, 2015:116-124.

[11] Martens J, Medabalimi V. On the expressive efficiency of sum product networks[J]. Computer Science, 2014, 1:102-110.

[12] Peharz R, Tschiatschek S, Pernkopf F, et al. On theoretical properties of sum-product networks[J]. Journal of Machine Learning Research, 2015, 38:744-752.

[13] Zhao H, Poupart P, Gordon G. A unified approach for learning the parameters of sum-product networks[C]// Proceedings of the 29th Advances in Neural Information Processing Systems, Barcelona, Spain: MIT Press, 2016, 12:146-153.

[14] Dennis A, Ventura D. Learning the architecture of sum-product networks using clustering on varibles[C]//Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, USA: MIT Press, 2012: 2033-2041.

[15] Adel T, Balduzzi D, Ghodsi A. Learning the structure of sum-product networks via an svd-based algorithm[C]// Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain: AUAI, 2015:32-41.

[16] Rashwan A, Zhao H, Poupart P. Online and distributed Bayesian moment matching for parameter learning in sum-product networks[C]//Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, Cadiz, Spain: JMLR, 2016:1469-1477.

[17] Nath A, Domingos P. Learning tractable probabilistic models for fault localization[C]//Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, New Orleans, LA, USA:AAAI, 2016:1294-1301.

[18] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.