

# 基于马尔科夫毯约束的最优贝叶斯 网络结构学习算法

谭翔元,高晓光,贺楚超

(西北工业大学电子信息学院,陕西西安 710129)

**摘要:** 本文针对最优贝叶斯网络的结构学习问题,在动态规划算法(Dynamic Programming, DP)的基础上,使用 IAMB 算法(Incremental Association Markov Blanket, IAMB)计算得到的马尔科夫毯对评分计算过程进行约束,减少了评分的计算次数,提出了基于马尔科夫毯约束的动态规划算法(Dynamic Programming Constrained with Markov Blanket, DPCMB),研究了 IAMB 算法中重要性阈值对 DPCMB 算法的各项性能指标的影响,给出了调整阈值的合理建议.实验结果表明,DPCMB 算法可以通过调整重要性阈值,使该算法的精度与 DP 算法相当,极大地减少了算法的运行时间、评分计算次数和所需存储空间.

**关键词:** 贝叶斯网络结构学习; 动态规划算法; 马尔科夫毯; IAMB 算法

**中图分类号:** TP18      **文献标识码:** A      **文章编号:** 0372-2112 (2019)09-1898-07

**电子学报 URL:** <http://www.ejournal.org.cn>      **DOI:** 10.3969/j.issn.0372-2112.2019.09.012

## Learning Optimal Bayesian Network Structure Constrained with Markov Blanket

TAN Xiang-yuan, GAO Xiao-guang, HE Chu-chao

(School of Electronic and Information, Northwestern Polytechnical University, Xi'an, Shaanxi 710129, China)

**Abstract:** To solve the problem about structure learning of optimal Bayesian network, this paper proposes dynamic programming constrained with Markov blanket (DPCMB), which uses Markov blanket calculated by incremental association Markov blanket (IAMB) to reduce the number of scoring calculations in dynamic programming. We research on the effect of the significance value in IAMB on the performance indicators of DPCMB algorithm, and give reasonable suggestions for adjusting the significance value. Experimental results show that the DPCMB algorithm can adjust the significance value so that the accuracy of the algorithm is comparable to that of the DP algorithm, and running time, score calculation times, and memory requirements of the algorithm are greatly reduced.

**Key words:** Bayesian network structure learning, dynamic programming, Markov blanket, IAMB

### 1 引言

贝叶斯网络(Bayesian Networks, BN)是一种表示随机变量间依赖和独立关系的网络模型,在目标识别、行为识别、医疗诊断、威胁评估<sup>[1]</sup>、敏感性分析等多个领域有着许多的应用,因此对贝叶斯网络的研究有着重要的意义.

贝叶斯网络的结构学习是利用相应的方法从样本数据集中确定贝叶斯网络节点之间的拓扑关系,然而,从所有可能的网络结构空间中搜索最优的贝叶斯网络

结构已经被证明是一个 NP-Hard (Non-deterministic Polynomial Hard, NP-Hard)问题<sup>[2]</sup>. 我们可以将贝叶斯网络结构学习的方法分为两大类:近似学习方法与精确学习方法. 近似学习方法只能做到局部最优或接近于最优网络结构<sup>[3]</sup>,并不能得到最优的贝叶斯网络结构. 随着研究与应用的发展,对贝叶斯网络结构学习的精度要求越来越高,于是相应的精确学习方法应运而生.

贝叶斯网络结构的精确学习有很多方面的研究,在众多已有文献中,基于动态规划的算法因其在中小规模的贝叶斯网络的结构学习中效果出众,故相关研究较多.

2004年,Koivisto与Sood提出了一种基于节点序的动态规划算法<sup>[4]</sup>,但是该动态规划算法的父节点的数量上限是人为给定的.2005年,Singh与Moore也提出了类似的动态规划算法<sup>[5]</sup>,且没有人为限定父节点的数量上限.2006年,Silander和Myllymaki提出了一种更为简洁的动态规划算法<sup>[6]</sup>(SMDP算法),再次减少了算法的运行时间,并且扩大了可计算的网路规模.2011年,Malone和Yuan提出了MEDP算法<sup>[7]</sup>(Memory-Efficient Dynamic Programming, MEDP),再次提升了算法的效率.他们利用评分函数的性质对父节点数量上限进行了约束,之后采用按层存储与计算的方式进行动态规划,极大地提高了效率.

之后,学习最优贝叶斯网络结构的算法在动态规划算法以外又有新的进展.

2011~2013年,Yuan和Malone基于最短路搜索问题的思想<sup>[8]</sup>,提出了A\*算法与AWA\*算法<sup>[9]</sup>(Anytime Window A\* algorithm, AWA\*).2012~2017年,Cussens等人<sup>[10]</sup>基于整数规划的思想,提出了GOBNILP系列算法(Globally Optimal Bayesian Network learning using Integer Linear Programming, GOBNILP).这些算法能够迅速得到网络结构,但是不能保证总是最优的,而使用动态规划算法可以学习得到最优的网络结构.由此可见,基于动态规划的算法在学习最优贝叶斯网络领域中始终占有一席之地.

上述动态规划算法,在无任何约束的情况下,算法的时间复杂度与空间复杂度均为 $O(n^2)$ ,除去Koivisto与Sood提出的动态规划算法与MEDP算法外,其余动态规划算法并未对搜索空间进行约束.这或许是近些年来动态规划算法没有实质性进展的原因之一.值得注意的是,Koivisto与Sood提出的动态规划算法是人为给予父节点上限,需要专家经验,而MEDP算法利用评分函数性质得到的约束条件不强,所以,如何通过引入更进一步的约束来提升动态规划算法效率是值得研究的.

本文通过引入马尔科夫毯这一更强的约束对动态规划算法的搜索空间进行更为严格的缩减,从而提升基于动态规划的贝叶斯网络的结构学习算法的效率.

## 2 动态规划算法原理及其分析

贝叶斯网络表示为 $BN = (G, P)$ ,  $G = (V, E)$ 为有向无环图,其中 $V = \{x_1, x_2, \dots, x_n\}$ 为节点集,与领域内的随机变量一一对应; $E$ 为有向边集,反映变量之间的因果依赖关系; $P$ 为节点的概率分布,表示节点之间的因果影响强度.在有向图中,如果从节点 $x_i$ 到节点 $x_j$ 有一条有向边 $x_i \rightarrow x_j$ ,则称 $x_i$ 为 $x_j$ 的父节点, $x_j$ 为 $x_i$ 的子节点,用 $Pa(x_i)$ 表示 $x_i$ 的所有父节点集合,用 $Ch(x_i)$ 表示 $x_i$ 的所有子节点集合.

基于动态规划的贝叶斯网络结构学习问题,就是根据完备数据集 $D$ ,能够寻找一个最优的贝叶斯网络结构 $G^*$ ,满足如下式子:

$$G^* \in \arg \max_G \text{Score}(G) \quad (1)$$

其中:

$$\text{Score}(G) = \sum_{x \in V} \text{LocalScore}(x | Pa(x)) \quad (2)$$

Score称为结构评分,LocalScore称为局部评分.大多数可分解的评分函数都可以作为Score,例如,BD(Bayesian Dirichlet, BD)评分、BIC(Bayesian Information Criterion, BIC)评分、MDL(Minimum Description Length, MDL)评分等等,当使用MDL评分时,所有相关式子应取最小值.

需要注意的是,最优的贝叶斯网络结构 $G^*$ 可能不是唯一的.多个拥有相同概率分布的贝叶斯网络,称为等价结构类,并且已经有证明<sup>[11]</sup>,互为等价结构类的贝叶斯网络在上述同一评分函数下具有相同的评分值.

动态规划算法的思想就是将搜索最优贝叶斯网络这一问题层层分解为寻找最优子网络,直至空节点,并递归求得最优网络.有核心递归公式:

$$\text{Score}(V) = \max_{x \in V} \{ \text{Score}(V \setminus \{x\}) + \text{BestScore}(x, V \setminus \{x\}) \} \quad (3)$$

$$\text{BestScore}(x, V \setminus \{x\}) = \max_{PS \subseteq V \setminus \{x\}} \text{LocalScore}(x | PS) \quad (4)$$

对于节点集 $V$ 的网络结构评分,可以分解为两部分:①基于节点集 $V \setminus \{x\}$ 的最优网络评分;②从节点集 $V \setminus \{x\}$ 中寻找 $x$ 的最优父节点集,使 $x$ 的局部评分最大.进而递归地,对于节点集 $V \setminus \{x\}$ 可以再次分解,直至分解为空集.于是,动态规划过程如下:从空集开始,找到单个变量的最优结构,并根据这些基本结构,为越来越大的变量集建立最佳子网络,直到找到最优网络为止.4节点的贝叶斯网络结构的动态规划算法过程见图1.已有证明<sup>[5,6]</sup>,动态规划算法的时间复杂度与空间复杂度均为 $O(n^2)$ .

上述即是动态规划算法的核心思路,各学者在具体的存储评分和搜索方式上进行了不同的探索与尝试,发展出了不同的动态规划算法,但是最本质的思路仍是相同的.

基于动态规划的贝叶斯网络结构学习算法需要将所有的局部评分 $\text{LocalScore}(x | PS)$ 提前计算好并存储<sup>[4-6]</sup>.对于 $n$ 个节点的贝叶斯网络,取任意节点 $x$ ,剩下的 $n-1$ 个节点可以组成 $2^{n-1}$ 个 $x$ 的可能父节点集合,总共需要 $n2^{n-1}$ 次局部评分计算,这也是最消耗时间与存储空间的部分.

有些学者提出了优化的方法,例如,Singh与Moore提出使用P-Cache存储结构来优化<sup>[5]</sup>,但是总共需要计

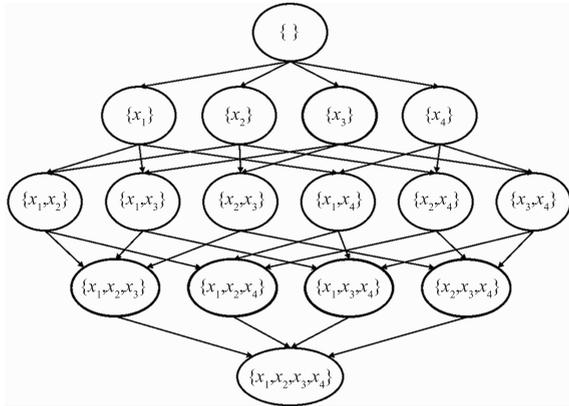


图1 4节点网络结构动态规划算法搜索图

算  $n2^{n-1}$  次局部评分的本质仍然没有改变. Koivisto 与 Sood 提出的动态规划算法人为约束父节点数量上限, 但是需要依靠专家经验. Malone 和 Yuan 提出的 MEDP 算法依靠 MDL 评分的性质<sup>[7]</sup>, 使父节点数量上限从  $n-1$  减少至  $m = \log(2N/\log N)$  ( $N$  为样本量), 仅仅是少量地减少了评分计算次数. 由此可见, 如何在动态规划算法中引入更强的约束进而提升效率是值得研究的.

### 3 基于马尔科夫毯约束的动态规划

在引出本文的基于马尔科夫毯约束的动态规划算法之前, 需要给出马尔科夫毯及其相关概念.

**定义 1** 阻塞: 通路  $\alpha$  被节点集  $Z$  阻塞当且仅当:

①  $\alpha$  包含一个顺连结构  $x_i \rightarrow x_k \rightarrow x_j$  或者包含一个分连结构  $x_i \leftarrow x_k \rightarrow x_j$ , 中间节点  $x_k$  属于节点集  $Z$ ; ②  $\alpha$  包含一个 V-结构  $x_i \rightarrow x_k \leftarrow x_j$ , 中间节点  $x_k$  及其子节点都不属于节点集  $Z$ .

**定义 2** d-分隔与条件独立: 在贝叶斯网络中, 如果节点  $x_i$  和  $x_j$  之间的所有通路均被节点集合  $Z$  所阻塞, 那么就说  $Z$  集合 d-分隔 (d-separate)  $x_i$  和  $x_j$ , 也可以说  $x_i$  和  $x_j$  在给定  $Z$  时条件独立, 表示为  $x_i \perp x_j \mid Z$  或  $I_D(x_i, x_j \mid Z)$ .

**定义 3** 马尔科夫毯: 设贝叶斯网络中一个节点  $x$ , 如果集合  $MB(x)$  可 d-分隔  $x$  和其余全部节点, 则称  $MB(x)$  是  $x$  的马尔科夫毯, 表示为:  $\forall Y \in V \setminus MB(x) \setminus \{x\}, I_D(x, Y \mid MB(x))$ .

在贝叶斯网络中, 马尔科夫毯  $MB(x)$  直观地表现为  $x$  的父子节点集与  $x$  的配偶节点集, 有:

$$MB(x) = Pa(x) \cup Ch(x) \cup \bigcup_{y \in Ch(x)} Pa(y) \quad (5)$$

这一性质将为后文对评分计算的约束起到至关重要的作用.

#### 3.1 基于马尔科夫毯约束的评分计算

对于式(4)在递归过程中的任意一次调用, 有:

$$BestScore(x, U) = \max_{PS \subseteq U} LocalScore(x \mid PS)$$

显然,  $U \subseteq V \setminus \{x\}$ , 而满足这样条件的  $PS \subseteq Pa(x)$ , 进而由式(5)可以得到:  $PS \subseteq Pa(x) \subset MB(x)$ . 又因为  $PS \subseteq U$ , 所以  $PS \subseteq U \cap MB(x)$ . 此外, 由马尔科夫毯的定义, 易知:  $MB(x) \subset V \setminus \{x\}$ , 那么, 对于  $x$  的所有的局部评分  $LocalScore(x \mid PS)$  而言, 可以得到:

$$PS \subseteq V \setminus \{x\} \cap MB(x) = MB(x)$$

上式的意义在于, 符合要求的所有  $PS$  仅在  $x$  的马尔科夫毯  $MB(x)$  中就可以找到, 不需要从  $x$  之外的所有节点集合  $V \setminus \{x\}$  中寻找. 于是, 可以得到最终约束后的  $x$  的所有的局部评分计算公式:

$$LocalScore(x \mid PS) \Rightarrow_{PS \subseteq V \setminus \{x\}} LocalScore(x \mid PS) \quad (6)$$

对于式(6)可以直观地理解为: 删减掉  $x$  基于马尔科夫毯  $MB(x)$  外的元素的局部评分的计算, 仅在马尔科夫毯中计算  $x$  基于其父节点集的评分, 从而减少了评分的计算次数, 将总体评分计算次数从  $n2^{n-1}$  减少至  $\sum_{x \in V} 2^{|MB(x)|}$ .

对于中等规模乃至大规模的贝叶斯网络, 这样的评分次数减少量是相当可观的. 试想, 如果让每个节点通过约束都删减掉一个可能的父节点集中的元素, 则计算评分的总次数减少了一半, 大大减少了不必要的评分计算与搜索.

那么, 通过此约束来减少评分计算次数的效果与 MEDP 算法通过 MDL 评分的性质减少评分计算次数的效果相比如何呢? 在此, 需要注意的是, 通过马尔科夫毯约束是对可能父节点集进行了删减, 将  $V \setminus \{x\}$  删减为  $MB(x)$ ; 而使用 MDL 评分的性质是对父节点数量上限的限制, 可能的父节点集仍是  $V \setminus \{x\}$ , 只是对于其中的子集  $PS \subseteq V \setminus \{x\}$ , 要求  $|PS| \leq m$ . 对于  $n$  个节点的贝叶斯网络, 需要将评分计算次数至少减少一半, 如果通过马尔科夫毯的约束, 只需要使每个节点的  $|MB(x)| \leq n-2$ , 即从每个节点的可能父集中删除至少 1 个不属于马尔科夫毯内的元素, 容易实现; 如果使用 MDL 评分的性质来减少, 总共需要  $n \sum_{i=0}^m C_{n-1}^i$  次评分计算, 有  $m = \log(2N/\log N) \leq \frac{n-1}{2}$ , 若令  $n=11$ , 则  $N \leq 500$ , 这样少的样本量是不足够对网络进行学习的, 故通过 MDL 评分的性质使评分次数减少一半是难以做到的, 所以通常情况下通过马尔科夫毯约束的方式减少得更多.

计算马尔科夫毯的算法有很多, 在众多算法中, 由于 IAMB 算法速度快<sup>[12]</sup>, 时间复杂度最低, 计算大样本数据时更有优势, 考虑到本文的目标是减少动态规划算法的时间, 故使用 IAMB 算法进行约束.

IAMB 算法可以通过输入数据集  $D$  与重要性阈值  $\epsilon$

得到目标变量  $x$  的马尔科夫毯  $MB(x)$ . 重要性阈值  $\varepsilon$  影响了最终计算出的马尔科夫毯的结果,  $\varepsilon$  越小, 结果越趋于严格, 所得的  $|MB(x)|$  越小; 反之,  $\varepsilon$  越大, 结果越趋于宽泛, 所得的  $|MB(x)|$  越大. 这一结果势必会影响之后的局部评分的计算, 进而影响整个动态规划过程, 更为详细的分析见第 4 节.

### 3.2 加入 IAMB 算法的局部评分计算

在计算了每个节点对应的马尔科夫毯后, 可以根据式 (6), 提前计算相应的约束后的全部局部评分  $LocalScore(x|PS)$ . 该过程见算法 1.

算法 1 基于 IAMB 约束的局部评分计算

```

计算马尔科夫毯约束下的所有局部评分
Function1  $P_{mask} = mkPossibleMask(MB)$ 
输入: MB. 由 IAMB 计算出的所有节点的马尔科夫毯
输出:  $P_{mask}$ .
01:  $P_{mask} \leftarrow all\ 0$  // 全置 0
02: for each  $x_i \in V$  do
03:   for each subset  $\subseteq MB(x_i)$  do
04:     code = SetToBinary(subset) // 根据集合生成对应的编码
05:      $P_{mask}[i][code + 1] = 1$ 
06:   end for
07: end for
Function2  $P_{score} = ComputeParScore(P_{mask}, D)$ 
输入:  $P_{mask}$ , 数据集  $D$ 
输出:  $P_{score}$ .  $n \times 2^n$  的矩阵, 对应  $P_{mask}$  存储评分
01:  $P_{score} \leftarrow all - Inf$  // 全置 - Inf
02: for each row of  $P_{mask}$  do
03:   PSCode = find( $P_{mask}[i][:] = 1$ )
04:   for each code of PSCode do
05:     PS = BinaryToSet(code) // 根据编码生成对应的集合
06:      $P_{score}[i][code + 1] = LocalScore(x_i | PS)$ 
07:   end for
08: end for

```

在该过程中, 先通过  $n \times 2^n$  大小的 0/1 矩阵  $P_{mask}$  记录对应编码位置的局部评分是否需要计算, 置 1 则对应编码位置局部评分计算, 置 0 则不计算.  $P_{mask}$  矩阵的第  $i$  行第  $j$  列 ( $j = code + 1$ , code 为 PS 的二进制编码) 的位置对应于局部评分  $LocalScore(x_i | PS)$ . 具体的 code 编码规则为: 根据  $x_n \rightarrow x_1$  的顺序, 对应位置父节点存在记作 1, 不存在记作 0. 例如, 以 4 节点的贝叶斯网络为例, 当需要对  $LocalScore(x | PS)$  进行计算时, 有  $i = 2$  且  $code = (0101)_2$ , 将  $P_{mask}$  的第 2 行第 6 列置 1, 表示需要计算对应的局部评分.

之后对经过马尔科夫毯约束后的  $P_{mask}$  矩阵, 产生  $P_{score}$  矩阵存储对应的局部评分.

通过上述过程, 能够将局部评分计算总次数从  $n2^{n-1}$  减少至  $\sum_{x \in V} 2^{|MB(x)|}$ . 当评分计算次数减少后, 会减

少对应的计算时间, 从而使整个算法的耗时减少, 也意味着算法需要存储的评分结果变少, 也减少了评分所占存储空间的大小.

最后, 一旦所有需要的局部评分被提前计算好, 就可以根据动态规划算法进行搜索. 相应的搜索过程原理见第 2 节, 具体执行仍按照 SMDP 算法<sup>[6]</sup> 的搜索过程执行, 该算法的搜索过程经过作者在已有的动态规划算法上进行了提高, 是相当简单、高效而又准确的.

由于本文的方法限制了评分的计算量, 使其仅在 I-AMB 算法得出的马尔科夫毯内计算评分, 而 IAMB 算法得出的结果又受其自身重要性阈值  $\varepsilon$  的影响, 可以预见, 本文提出的 DPCMB 算法学习得到的贝叶斯网络结果会随着  $\varepsilon$  的取值而产生变化, 详细说明见第 4 节.

## 4 实验及分析

本文运行贝叶斯网络结构学习算法的实验环境是 Windows 10, Intel Core i7-7700 CPU @ 3.60GHz, RAM 8.00GB, 使用 Matlab 2014x64 进行实验.

### 4.1 重要性阈值对算法影响的实验及分析

在第 3.1 节中初步分析可知,  $\varepsilon$  值会影响本文提出的 DPCMB 算法的各项性能. 在此, 将研究  $\varepsilon$  值对 DPCMB 算法的运行时间、评分函数计算次数、精度和所占存储空间大小的影响.

需要说明的是, 为了回避遍历等价结构类的问题, 本文在评估算法的学习精度时, 将通过算法计算出的网络结构评分与原始网络结构评分误差的百分比来衡量, 用公式表述为:

$$Dev = \left| \frac{Score(G') - Score(G)}{Score(G)} \right| \times 100\% \quad (7)$$

其中,  $G'$  表示用算法学习出来的网络结构,  $G$  表示原始网络结构, Score 为评分函数, 在本文中统一使用 BIC 评分, 后文均用此衡量算法精度. 由式 (7) 可知, 评分误差百分比越小, 算法的精度越高.

通过多次实验, 本文提出的 DPCMB 算法在诸多不同的贝叶斯网络下, 运行时间、评分函数计算次数、精度和所占存储空间大小四方面都保持了相似的趋势, 故只选取算法在某一特定贝叶斯网络 (节点数: 13, 有向边数: 21, 参数量: 262) 中的具体数据加以说明, 如图 2 所示.

如图 2 中, 使用 DPCMB 算法对贝叶斯网络的结构进行学习, 算法的运行时间、评分函数的计算次数与所占存储空间随着重要性阈值  $\varepsilon$  的增加而迅速增长; 由式 (7) 计算出的评分误差百分比随着  $\varepsilon$  趋于 1 而逐步下降, 最终稳定趋近于 0, 也由此可见, 精度与时间是算法中不可兼得的两项指标.

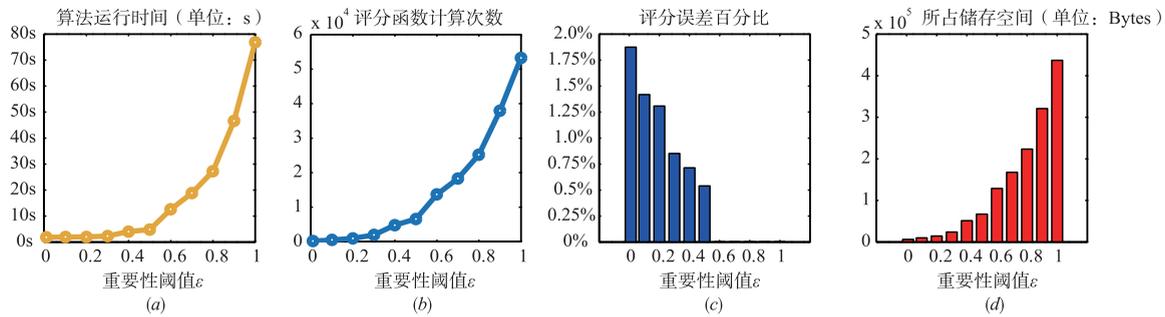


图2 DPCMB算法在13节点网络中各项指标变化

结合 3.1 节的分析,本文提出的 DPCMB 算法,在重要性阈值  $\epsilon$  趋近于 0 时,计算所得马尔科夫毯的结果更为严格,  $|MB(x)|$  小,由  $\sum_{x \in V} 2^{|MB(x)|}$  可知计算评分函数的次数更少,相应的运算时间减少,且存储评分所需空间减少,另一方面,所得马尔科夫毯的结果严格,意味着排除的节点较多,本应该计算、搜索的评分排除较多,算法精度下降;反之则反.实际上,当  $\epsilon$  趋近于 1 时,  $|MB(x)| \rightarrow n-1$ ,由  $\sum_{x \in V} 2^{|MB(x)|}$  可知评分函数的计算次数更接近于  $n2^{n-1}$ ,算法的效率更趋近于 SMDP 算法.特别地,当  $\epsilon=1$  时,本文提出的 DPCMB 算法理论上等价于 SMDP 算法.

值得注意的是,本文的算法并非仅在  $\epsilon=1$  时评分误差百分比才趋近于 0,而是早在  $\epsilon=1$  之前,算法的评分误差百分比就稳定地趋近于 0. 在 13 个节点的贝叶斯网络中,该现象在  $0.5 < \epsilon < 0.6$  就已经发生了,而此时的 DPCMB 算法运行时间、评分函数计算次数和所占存储空间大小仍是远远小于 SMDP 算法学习该网络时的各项指标.

于是,可得出结论,DPCMB 算法可以通过调整  $\epsilon$  使其精度与 SMDP 算法相当,同时使时间、评分计算次数和存储空间小于 SMDP 算法,从而获得比 SMDP 算法更加良好的效果.

此外,根据上述规律,对于本文提出的算法,可以给出如下的调整参数建议:在追求较高学习精度的前提下,将重要性阈值  $\epsilon$  适量调大;在追求较高学习速度的前提下,将重要性阈值  $\epsilon$  适量调小,具体调整需要根据实际应用情况进行实验.

#### 4.2 与其他动态规划算法实验对比及分析

此处将 DPCMB 算法与 SMDP 算法、MEDP 算法进行运行时间、评分计算次数、精度和所占存储空间大小的对比.本文中用来进行结构学习的贝叶斯网络的节点个数  $n$ 、有向边条数  $E$ 、参数量  $P$  见表格,样本量  $N=10000$ ,所得结果如表 1 所示(此处 DPCMB 算法的重要性阈值  $\epsilon=0.50$ ).

表 1 三种 DP 算法的运行时间与评分函数计算次数的比较

网络参数			运行时间(sec)			评分计算次数			评分误差百分比(%)			存储空间大小(bytes)		
$n$	$E$	$P$	SMDP	MEDP	DPCMB	SMDP	MEDP	DPCMB	SMDP	MEDP	DPCMB	SMDP	MEDP	DPCMB
10	14	140	1	2	1	5120	5020	2400	4.18E-13	4.77E-13	4.18E-13	4.24E+4	1.57E+4	2.15E+4
11	17	178	3	4	1	11264	10648	2704	2.04E-12	1.09E-12	2.04E-12	9.32E+4	2.88E+4	2.54E+4
11	17	370	15	7	3	11264	10648	3712	3.60E-12	3.60E-12	3.60E-12	9.36E+4	2.91E+4	3.48E+4
11	17	548	28	10	2	11264	10648	2360	2.06E-12	8.81E-13	0.12	9.39E+4	2.83E+4	1.45E+4
12	20	53	3	7	1	24576	21792	5504	0.0040	0.0040	0.3510	2.01E+5	6.13E+4	5.10E+4
12	20	280	14	10	2	24576	21792	3488	7.57E-13	2.38E-12	0.78	2.02E+5	6.01E+4	3.57E+4
12	20	480	14	11	2	24576	21792	5376	8.21E-13	2.49E-13	8.21E-13	2.03E+5	5.63E+4	5.18E+4
13	21	64	6	15	2	53248	42926	5540	0.0019	0.0029	0.0019	4.35E+5	1.14E+5	5.65E+4
13	21	262	66	25	5	53248	42926	6560	1.56E-13	3.23E-12	0.5296	4.37E+5	1.08E+5	6.69E+4
13	21	553	359	41	4	53248	42926	4160	0.16	2.39E-12	0.16	4.41E+5	1.08E+5	6.36E+4
14	23	102	30	39	4	114688	81368	10240	1.08E-12	6.64E-13	0.3085	9.47E+5	2.26E+5	1.09E+5
14	21	288	301	57	7	114688	81368	8400	4.36E-14	1.85E-12	4.36E-14	9.42E+5	2.14E+5	1.93E+5
15	24	111	98	97	17	245760	148620	34816	0.0069	0.0069	0.0069	2.03E+6	4.31E+5	3.53E+5
15	30	302	338	112	27	245760	148620	37184	1.62E-13	6.94E-14	1.02	2.03E+6	4.67E+5	3.72E+5
16	23	131	611	318	25	524288	262144	36200	1.28E-12	2.39E-13	1.28E-12	4.34E+6	8.59E+5	3.90E+5
17	24	153	1958	1163	46	1114112	447661	43072	0.0020	0.0020	0.0020	9.21E+6	1.65E+6	5.36E+5
18	26	167	10284	5610	327	2359296	742068	127232	1.13E-13	9.70E-13	0.2098	1.94E+7	3.48E+6	1.48E+6
19	27	172	31954	24564	350	4980736	1197076	190464	1.59E-12	2.21E-12	1.59E-12	4.13E+7	6.90E+6	2.41E+6
20	25	230	106078	93175	530	10485760	1883680	177472	2.84E-4	6.65E-13	2.84E-4	8.67E+7	1.34E+7	2.82E+6

如表 1 所示,对于相同贝叶斯网络结构的学习,SMDP 算法用时最多,MEDP 算法用时次之,本文提出的 DPCMB 算法用时最少,且在评分计算次数和所占存储空间大小上有相同的趋势.对于相同节点数的网络的学习,SMDP 算法与 MEDP 算法有着恒定的评分计算次数.在算法精度上,除去灰色标注部分的数据,DPCMB 算法与 SMDP 算法的评分误差百分比一致,与 MEDP 算法的评分误差百分比不同,但是三者均接近于 0.

由 3.1 节可知,对于 SMDP 算法,评分计算次数是  $n2^{n-1}$ ,只受到节点数  $n$  的影响,故在相同节点数的网络下评分计算次数是恒定的;对于 MEDP 算法,总共需要  $n \sum_{i=0}^m C_{n-1}^i$  次评分计算,只受节点数  $n$  与样本量  $N$  影响,故在相同节点数的网络下(样本量均为 10000)评分计算次数也是恒定的,且通常情况下  $m \leq n-1$ ,于是有  $n \sum_{i=0}^m C_{n-1}^i \leq n2^{n-1}$ ,故对于同一贝叶斯网络的结构学习,MEDP 算法评分计算次数少于 SMDP 算法的次数;对于 DPCMB 算法,评分计算次数为  $\sum_{x \in V} 2^{|\text{MB}(x)|}$ ,由于网络结构的不同会得到不同的  $\text{MB}(x)$ ,故总的评分计算次数不是恒定的,但是评分计算次数是最少的.正是由于在评分计算次数这一指标上,SMDP 算法最多,MEDP 算法次之,DPCMB 算法最少,所以在与评分计算次数正相关的算法运行时间与所占存储空间大小两项指标上,三种算法有着相同的关系.

由于 DPCMB 算法的搜索部分仍是采用 SMDP 算法的搜索方式,故当 DPCMB 算法精度逐步提高时,最终与 SMDP 算法的精度一致是在意料之中;MEDP 算法则大幅度改进了存储与搜索方式,结果与二者不同,但是同属于精确学习方法,故评分误差百分比都趋近于 0.除去灰色标注部分,DPCMB 算法精度与 SMDP 算法一致,说明本文提出的算法的确能在一定程度内既能够减少计算、存储不必要的评分,又能够维持自身算法的精度.

总而言之,三种动态规划算法的评分计算次数理论最大值均是  $n2^{n-1}$ ,SMDP 算法自身并未约束,MEDP 算法进行了少量的缩减,本文的 DPCMB 算法在  $\varepsilon = 0.50$  的情况下减少计算量在三者中最多,故三种算法的评分计算次数依次递减,运行时间和所占存储空间大小也依次递减.

综上,本文提出的 DPCMB 算法对比前两种动态规划算法在运行时间、评分计算次数和所占存储空间大小三方面有着相当明显的优势.

对于灰色标注的部分,本文的 DPCMB 算法精度变低,未达到与 SMDP 算法精度一致并接近于 0 的情况,再单独研究,进行实验,逐步提高  $\varepsilon$  值使精度逼近 SMDP 算

法的精度,记录各项指标,所得结果如表 2 所示.

表 2 DPCMB 算法在灰色标注网络中各项指标

网络参数			运行时间(s)	评分计算次数	评分误差百分比(%)	存储空间大小(byte)	$\varepsilon$
$n$	$E$	$P$					
11	17	548	2	2208	2.06E-12	2.23E+4	0.60
12	20	53	2	5696	0.0040	5.28E+4	0.53
12	20	280	2	4896	1.81E-12	4.82E+4	0.58
13	21	262	9	6464	1.56E-13	8.95E+4	0.55
14	23	102	8	22016	1.08E-12	2.07E+5	0.55
15	30	302	56	63232	1.62E-13	6.04E+5	0.60
18	26	167	544	345088	1.13E-13	3.49E+6	0.60

如表 2 所示,通过将重要性阈值  $\varepsilon$  进一步调高后,DPCMB 算法的评分误差百分比大多数提高到了与 SMDP 算法所得评分误差百分比相同的地步.这也与 4.1 节的结论相互印证,即:当  $\varepsilon$  逐渐趋近于 1 时,DPCMB 算法的精度趋近于 SMDP 算法,并最终趋于稳定.

此外,将表 2 所得结果与表 1 对应网络所得结果相对比,提高重要性阈值  $\varepsilon$  后的 DPCMB 算法的运行时间、评分函数的计算次数与所占存储空间三项指标均略有增加,但是仍少于 SMDP 算法与 MEDP 算法所对应指标.综上所述,本文提出的 DPCMB 算法可以通过适当调控  $\varepsilon$ ,使精度保持不变,所需时间、评分计算次数和所需存储空间得到极大的减少,这就是本文提出的算法相较于其他现有优秀的动态规划算法的优势.

最后,对于每个特定的贝叶斯网络,适合达到与 SMDP 算法精度一致的  $\varepsilon$  值是不同的.随着所学习的贝叶斯网络变得复杂,运行算法的时间也会随之增加,对  $\varepsilon$  进行调整的难度会进一步增加.因此,如果并不过分追求结构学习的精度,反而可以适当减少  $\varepsilon$ ,从而获得运行时间与所需存储空间的减少,以提高算法效率.

## 5 总结

本文对动态规划算法的搜索空间进行了约束,使局部评分计算仅在 IAMB 算法得到的马尔科夫毯范围内进行,删除了节点基于马尔科夫毯之外元素的局部评分的计算,提出了马尔科夫毯约束的动态规划算法,即 DPCMB 算法.该算法通过约束减少了局部评分的计算,进而减少了算法所需的运行时间和所需存储空间,而算法精度则取决于重要性阈值  $\varepsilon$ .实验结果表明,增大重要性阈值  $\varepsilon$ ,DPCMB 算法的运行时间、评分的计算次数与所占存储空间也会增加,而评分误差百分比会逐步下降(即精度逐步上升),最终趋近于 SMDP 算法的对应值.对相同的贝叶斯网络进行结构学习,相比已有文献中表现良好的两种动态规划算法,本文提出的 DPCMB 算法可以通过调整重要性阈值  $\varepsilon$  到某一特定值(视不同网络结构而不同),从而使算法的精度与 SMDP 算法、MEDP 算法的精度相当,而算法所需的运行时间、

评分计算次数和所需存储空间可以比 SMDP 算法、MEDP 算法更少。

最后,结合重要性阈值  $\varepsilon$  对算法四种性能指标的影响规律,我们可以给出使用建议:在追求贝叶斯网络结构学习精度的前提下,将重要性阈值  $\varepsilon$  向趋近于 1 的方向调整;在追求贝叶斯网络结构学习速度的前提下,将重要性阈值  $\varepsilon$  向趋近于 0 的方向调整。

本文只得到了重要性阈  $\varepsilon$  值与四种指标的大致关系,对于满足需求精度的  $\varepsilon$  值选取只能沿大致趋势进行调整,而具体如何有效地调整  $\varepsilon$  值以满足需求精度的方法仍需要今后的进一步研究。

#### 参考文献

- [1] 邸若海,高晓光,郭志高. 小数据集 BN 建模方法及其在威胁评估中的应用[J]. 电子学报,2016,44(6):1505-1511.  
DI Ruo-hai, GAO Xiao-guang, GUO Zhi-gao. The modeling method with Bayesian networks and its application in the threat assessment under small data sets[J]. Acta Electronica Sinica,2016,44(6):1505-1511. (in Chinese)
- [2] CHICKERING DM, HECKERMAN D, MEEK C. Large-sample learning of Bayesian networks is NP-hard[J]. Journal of Machine Learning Research, 2004, 5(10):1287-1330.
- [3] 朱明敏,刘三阳,杨有龙. 基于混合方式的贝叶斯网络等价类学习算法[J]. 电子学报,2013,41(1):98-104.  
ZHU Ming-min, LIU San-yang, YANG You-long. Structural learning Bayesian network equivalence classes based on a hybrid method[J]. Acta Electronica Sinica,2013,41(1):98-104. (in Chinese)
- [4] KOIVISTO M, SOOD K. Exact Bayesian structure discovery in Bayesian networks[J]. Journal of Machine Learning Research, 2004, 5(5):549-573.
- [5] SINGH A, MOORE A. Finding optimal Bayesian networks by dynamic programming[R]. Pittsburgh: Carnegie Mellon University, 2005.
- [6] SILANDER T, MYLLYMAKI P. A simple approach for finding the globally optimal Bayesian network structure [A]. Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence [C]. Arlington: AUAI Press, 2006. 445-452.
- [7] MALONE B, Yuan C. Memory-efficient dynamic programming for learning optimal Bayesian networks[A]. Proceedings of 25th AAAI Conference on Artificial Intelligence [C]. Palo Alto: AAAI Press, 2011. 1057-1062.
- [8] YUAN C, MALONE B. Learning optimal Bayesian networks: a shortest path perspective[J]. Journal of Artificial Intelligence Research, 2013, 48(1):23-65.
- [9] MALONE B, YUAN C. Evaluating anytime algorithms for learning optimal Bayesian networks[A]. Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence [C]. Corvallis: AUAI Press, 2013. 381-390.
- [10] BARTLETT M, CUSSENS J. Integer linear programming for the Bayesian network structure learning problem[J]. Artificial Intelligence, 2017, 244(244):258-271.
- [11] CHICKERING DM. A transformational characterization of equivalent Bayesian network structures [A]. Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence [C]. San Francisco: Morgan Kaufmann, 1995. 87-98.
- [12] TSAMARDINOS I, ALIFERIS C F. Algorithms for large scale Markov blanket discovery [A]. Proceedings of the 16th International FLAIRS Conference [C]. Palo Alto: AAAI Press, 2003. 376-381.

#### 作者简介



谭翔元 男,1995年6月出生于四川遂宁. 现为西北工业大学博士研究生. 主要研究方向为贝叶斯网络结构学习.  
E-mail: tanxy2017@mail.nwpu.edu.cn



高晓光(通信作者) 女,1957年出生于辽宁沈阳. 现为西北工业大学教授,博士生导师. 主要研究方向为复杂系统建模及效能评估.  
E-mail: cxg2012@nwpu.edu.cn



贺楚超 男,1992年3月出生于陕西西安. 现为西北工业大学博士研究生. 主要研究方向为贝叶斯网络学习与应用.  
E-mail: xomrssh@163.com