

一种优化节点序搜索算子的BN结构学习方法

贾柳娜,董绵绵,贺楚超,邸若海,李晓艳

(西安工业大学 电子信息工程学院,陕西 西安 710021)

摘要:节点序空间下的局部搜索算法是一种性能良好的贝叶斯网络结构学习方法,在学习效率上具有极大的优势。然而,现有的该类算法通常存在节点序优化不足、学习精度低等问题,并容易停止在一个局部最优。为解决这些问题,对节点序空间下的局部搜索算法进行了研究,提出了一种新的通过优化节点序搜索算子来提高贝叶斯网络结构学习精度的IWINOBS算法。将迭代局部搜索算法与窗口算子相结合来搜索节点序空间中给定节点序的邻域,减小了算法陷入局部最优值的概率,从而获得质量更高的网络结构。实验结果表明:与网络结构空间下的贝叶斯网络结构学习算法相比,所提算法的学习效率提升了54.12%;与现有节点序空间下的贝叶斯网络结构学习算法相比,所提算法的学习精度提高了2.33%。

关键词:贝叶斯网络;结构学习;节点序优化;搜索算子;局部搜索

中图分类号:TP181

文献标志码:A

文章编号:1000-2758(2023)02-0419-09

贝叶斯网络(Bayesian networks, BN)^[1]是一种用于描述随机变量之间因果关系的概率图形模型,能有效直观地表达和推理不确定性问题。现如今各领域数据呈现出爆炸式的增长趋势,传统的基于专家经验的决策方法已经不能满足社会发展的需要。而贝叶斯网络在处理不确定性问题上的巨大优势使其广泛应用于数据挖掘、机器学习、诊断和评估等领域^[2]。在贝叶斯网络学习中,贝叶斯网络结构学习是参数学习和推理的前提和基础,也是贝叶斯网络研究的重点。因此,如何在合理的时间范围内从复杂的数据中学习最优结构近年来引起了国内外专家学者的大量研究关注。

基于评分搜索的方法是一种常用的BN结构学习算法^[3],通过优化评分函数或搜索策略来改善学习效率,其中常用的评分函数包括BIC^[4]、MDL^[5]和BD^[6]评分等。评分搜索法的搜索空间的可分为网络结构空间^[7]和节点序空间^[8]两大类。典型的网络结构空间下的BN结构学习算法包括K2算法^[9]、HC算法^[10]等,该类算法的复杂度随着网络中节点数目的增加呈指数级增长^[11],搜索效率较低。本文

重点研究的是节点序空间下基于评分搜索的贝叶斯网络结构学习问题,相较于网络结构空间,基于节点序空间(ordering-based search, OBS)展开搜索时有以下优势^[12]:①搜索范围由网络空间下的 $2^{\Omega(n^2)}$ 个候选网络结构转变为节点序空间下的 $2^{O(n \lg n)}$ 条节点序,搜索空间明显缩小;②基于网络空间展开搜索时每一步的构造网络结构和执行条件独立性测试都很耗时,而节点序空间下进行搜索无需构造网络结构,搜索效率更高;③搜索过程中的每一步对当前假设进行的全局性修改程度更大,能够有效避免算法陷入局部最优。

目前,国内外专家学者已提出许多通过搜索最优节点序来学习评分最高的BN结构方法,代表性的方法有OBS^[8]、INOBS^[13]、IINOBS^[13]等,通过搜索最优节点序来学习评分最优的贝叶斯网络结构。这些节点序空间下的BN结构学习算法均取得了较好的学习效果,然而在展开搜索时仍然存在节点序优化不足、学习精度低等问题。为了解决这些问题,本文提出了一种新的基于节点序的BN结构学习算法,通过优化节点序搜索算子使当前节点的排序发

收稿日期:2022-06-23

基金项目:陕西省自然科学基金基础研究计划(2022JQ-590)资助

作者简介:贾柳娜(1997—),西安工业大学硕士研究生,主要从事机器学习及贝叶斯网络结构学习研究。

通信作者:董绵绵(1981—),西安工业大学副教授,主要从事无线通信与信息处理研究。e-mail:dong_mm@aliyun.com

生局部变化的程度更大,从而获得更高评分的结构,将窗口算子与 INOBS 算法(insert neighbourhood OBS, INOBS)相结合作为迭代局部搜索过程的组成部分,提出基于迭代局部搜索的引入窗口算子的 INOBS 算法(iterate window operator with INOBS, IWINOBS)。仿真结果表明,在中小规模网络下 IWINOBS 算法的学习效率和精度优于 OBS 和 IINOBS 算法等节点序空间下的经典算法。

1 贝叶斯网络结构学习

一个贝叶斯网络用 $B = (G, P)$ 表示,其中 $G(V, E)$ 表示贝叶斯网络结构,是一个有向无环图(directed acyclic graph, DAG), $V = \{X_1, X_2, \dots, X_n\}$ 表示随机变量的集合, E 表示有向边的集合,可以定义为每个节点的父节点集合 $P_a(X_i)$; P 是贝叶斯网络参数,表示每个变量在给定其父节点时的条件概率分布,定量地描述了变量与其父节点之间的因果依赖关系。图 1 给出了一个典型的贝叶斯网络模型,该网络模型包含 5 个节点和 4 条有向边,可以看到每个节点对应的条件概率分布表和节点之间的相关性。模型中所有节点的类型都是离散的,并且均含有 2 个状态值(y 和 n)。

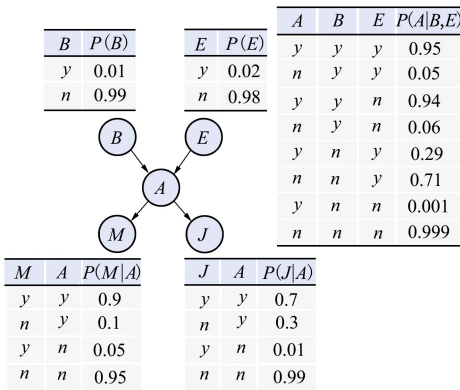


图 1 一个贝叶斯网络模型

由于局部马尔科夫性,即给定贝叶斯网络中节点 X_i 的父节点,则节点 X_i 条件独立于其所有的非后代节点。因此,完整联合概率分布表 $P(V)$ 可以表示为局部分布函数的乘积,大大降低了联合概率的计算复杂度。联合概率分布 $P(V)$ 的表达式如公式(1)所示,其中 $P_a(X_i)$ 表示节点 X_i 的父节点集合。

$$P(V) = P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | P_a(X_i)) \quad (1)$$

结构学习的目标是找到一个评分最高的有向无环图 G_* , 本文从完整数据集 $D = \{D_1, D_2, \dots, D_n\}$ 中学习贝叶斯网络结构,即 $G_* = \arg \max_{H \in G} (G, D)$, 其中, H 是所有可能的 DAG 的集合, s 表示评分函数,用来衡量网络结构的优劣。大多数评分函数都具有可分解性,即一个网络结构 G 的评分 $s(G, D)$ 可以用 G 中各个节点的局部评分之和来表示,因此很大程度上降低了在搜索过程中计算网络结构评分的复杂度,如公式(2)所示。

$$s(G, D) = \sum_{i=1}^n S(X_i | P_a(X_i)) \quad (2)$$

由公式(2)可知,对于结构 G , 其评分 $s(G, D)$ 仅依赖于节点 X_i 和 X_i 的父节点集 $P_a(X_i)$ 。

本文采用 BIC 作为评分函数,它与 DAG 的后验概率成正比。BIC 评分是可分解的,可以表示成各个节点的评分之和,如公式(3)所示。

$$s_{BIC}(G) = \sum_{i=1}^n s_{BIC}(X_i, \Pi_i) = \sum_{i=1}^n \sum_{\pi \in |\Pi_i|} (N_{x,\pi} \ln \hat{\theta}_{x|\pi}) - \frac{\ln N}{2} (|X_i| - 1) (|\Pi_i|) \quad (3)$$

公式(3)中,前一项是结构 G 的优参数似然度,用来衡量数据 D 与结构 G 的拟合程度;后一项添加了关于模型复杂性的惩罚项,从而有效避免过拟合。其中, $\hat{\theta}_{x|\pi}$ 是条件概率 $P(X_i = x | \Pi_i = \pi)$ 的最大似然估计, $N_{x,\pi}$ 表示 $X_i = x \wedge \Pi_i = \pi$ 在数据集中出现的次数, $|\cdot|$ 表示节点的笛卡尔积空间的大小, $|X_i|$ 是 X_i 的取值状态数且 $|\phi| = 1$ 。利用 BIC 评分的可分解性,可以独立地识别每个节点的候选父节点集列表 C_i , C_i 称为节点 X_i 的父节点集缓存。

2 IWINOBS 算法研究

基于节点序的 OBS 算法将 BN 结构学习问题转化为搜索最优节点序的问题,可以显著缩小搜索空间^[12]。给定节点序 \prec , 关于该节点序的最佳网络结构可以在 $O(Ck)$ 时间内找到,其中, $C = \sum_{i=1}^n |C_i|$, k 表示最大入度,节点 X_i 的最优父节点集为

$$P_{a\prec}(X_i) = \arg \max_{U \in U_i} s(X_i, U) \quad (4)$$

式中, U_i 是所有可能的父节点的集合,仅包含节点

序 < 中排在 X_i 之前的节点,这也被称为一致性规则。对节点序进行抽样后,就可以根据公式(4)快速获得给定节点序的最高评分结构。一般来说,在节点序空间中进行局部搜索的过程包括以下 4 个步骤^[13]:①采用启发式搜索随机获取初始节点序;②获得初始节点序后,使用不同的搜索算子对当前节点序的邻域展开搜索,以此来改变节点序中一些节点的位置,节点序每改变一次(迭代)说明已进行一次优化;③当一次迭代完成(未达到终止条件)时,通过更大的搜索步骤来优化当前节点序,从而避免陷入局部最优状态;④获得最优节点序后,再将该节点序作为 K2 算法的输入得到 DAG 结构。

2.1 节点序空间中的局部搜索算子

OBS 算法使用交换相邻算子对节点序进行搜索,交换相邻算子的定义如公式(5)所示,示例如图 2 所示。

$$O_{\text{swap}}(i) : (X_1, \dots, X_i, X_{i+1}, \dots, X_n) \rightarrow (X_1, \dots, X_{i+1}, X_i, \dots, X_n) \quad (5)$$

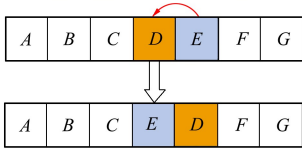


图 2 节点 E 和 D 在节点序中交换位置

由公式(5)可知,根据新的节点序来计算评分最高的网络的评分,只需要重新计算 X_i 和 X_{i+1} 的父节点集合。因此,给定节点序,使用一次交换相邻算子会产生 $n - 1$ 个候选节点序。虽然交换相邻算子简单有效,但其搜索空间有限,往往不足以获得最优节点序,这限制了它避免陷入局部最优值的能力。

文献[14]提到的 INOBS 算法中引入了插入算子,即给定 2 个索引 i 和 j ,初始节点序中索引 i 处的节点被插入后续节点序中的索引 j 处,使得排在初始节点序中索引 i 和 j 之间的节点位置发生变化。因此,使用插入算子对节点序的修改程度比交换相邻算子更大,从而克服了 OBS 算法的不足。插入算子的定义如公式(6)所示,示例如图 3 所示。

$$O_{\text{insert}}(i, j) : (X_1, \dots, X_i, \dots, X_j, \dots, X_n) \rightarrow (X_1, \dots, X_j, X_i, \dots, X_n) \quad (6)$$

由此可见,交换相邻算子也是插入算子的一种特殊情况,它应用于 2 个相邻的节点。也就是说 $O_{\text{swap}}(i) = O_{\text{insert}}(i + 1, i)$ 。

插入算子可以被看作是多次相邻节点交换操作

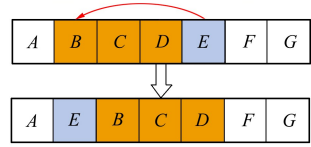


图 3 节点 E 被插入到节点 B 的位置

的结果,如图 4 所示。每次使用交换相邻算子只需要重新计算被交换的 2 个节点的父节点集合,因此,插入算子也可以只重新计算有限数量的父节点集合。

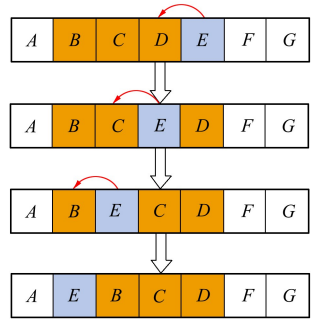


图 4 插入算子的具体执行方式

给定节点序,使用一次插入算子会产生 $n(n - 1)$ 个候选节点序。Alonso-Barba 等^[15]提出了一种搜索候选节点序的方法:随机选择一个节点作为固定索引,通过一系列交换算子完成对第二个索引的完整搜索,然后选择评分更优的候选节点序,直到所有节点都没有被进一步优化时停止搜索。

Scanagatta 等^[16]提出了一种用于局部搜索的窗口算子,窗口算子是插入算子的一种,包括起始位置 i 、插入位置 j 和组合节点的窗口大小 w 这 3 个参数,用 $O_{\text{window}}(i, j, w)$ 表示。其工作原理如下:选择初始节点序中位置 i 处的节点和 $w - 1$ 个后继节点,将这些节点插入到位置 j ,从而得到新的节点序。

窗口算子在另一个节点的前面插入多个节点,通过同时改变多个节点的索引来更新节点序,所有改变的节点必须是相邻的,因此局部搜索的范围会进一步扩大。窗口算子也可以通过一系列插入算子来实现,其规则如公式(7)所示,示例如图 5 所示。

$$O_{\text{window}}(i, j, w) : (X_1, \dots, X_j, \dots, X_i, \dots, X_{i+w}, \dots, X_n) \rightarrow (X_1, \dots, X_i, \dots, X_{i+w}, X_j, \dots, X_n) \quad (7)$$

由图 5 可知,给定节点序,使用一次窗口算子会产生 $(n - (w - 1)) \cdot (n - w)$ 个候选节点序。窗口

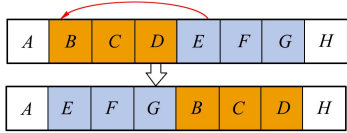


图 5 $O_{window}(5,2,3)$ 示例:节点 E, F, G 插入到初始节点序中节点 B, C, D 的位置

算子可以看作是一系列“窗口交换”,每次窗口交换最多需要重新计算与 $w + 1$ 个节点序一致的最优父节点集。窗口算子在节点序中的具体使用过程:选择一个随机节点作为固定索引,从 $w = 1$ 开始,考虑窗口大小为 w 的所有后继节点序;如果找到了评分更优的后继节点序,则移动到该状态并使得 $w = 1$;如果所有节点都已选择,并且没有发现可以提高评分的后继状态,则将 w 增加 1;如果超过了最大窗口大小,则返回当前状态。

2.2 节点序的迭代局部搜索方法

为提高节点序空间下 BN 结构学习算法的学习精度,本文在搜索节点序的过程中提出将迭代局部搜索(iterative local search, ILS)方法与窗口插入算子相结合的搜索方法。ILS 算法是一种具有 2 个算子的元启发式方法,用于提高局部最优解的质量。第一个是局部搜索算子,它能在一个解的邻域中找出局部最优解,第二个是扰动算子,能够在一个局部最优解的附近继续搜索,为局部搜索找到一个新的最优解。一般来说,迭代局部搜索方法由局部搜索、扰动和停止标准 3 个步骤组成。本文将窗口算子与迭代局部搜索相结合,从预先定义的搜索空间内的随机初始解开始,当前达到的局部最优值能够被随机交换扰动,得到扰动后的新解,通过窗口算子在扰动解的附近以更大的搜索步骤继续搜索更优节点序。若当前解大于扰动得到的解,返回当前解;否则,返回到前一步继续进行扰动交换,直到满足指定的终止条件。结合窗口算子后的迭代局部搜索算法图解如图 6 所示。

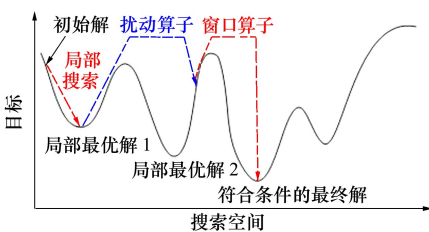


图 6 结合窗口算子后的迭代局部搜索算法图解

2.3 IWINOBS 算法构建

本文的方法扩展了迭代局部搜索的思想,采用增加窗口算子的窗口大小的方式对其进行优化。IWINOBS 算法的最大窗口大小从 $w = 1$ 开始,算法描述如下:

- 步骤 1 对初始节点序 O 进行局部搜索得到一个局部最优值 O' ;
- 步骤 2 将固定的迭代次数 P 应用于局部最优节点序 O' ,得到新解 O'' ;
- 步骤 3 使用窗口大小为 w 的窗口算子以更大的搜索步骤在节点序 O'' 附近继续搜索,得到节点序 O''' ;
- 步骤 4 若节点序 O''' 没有满足终止条件,则返回步骤 2 继续应用迭代局部搜索。每次迭代时再次应用窗口算子,并使窗口大小 $w = w + 1$,直到满足指定的终止条件时停止搜索。

给定初始节点序时,窗口大小 w 既影响算子避免局部最优值的能力,也会影响计算所需的时间。

IWINOBS 算法流程图如图 7 所示,算法的伪代码如下所示。

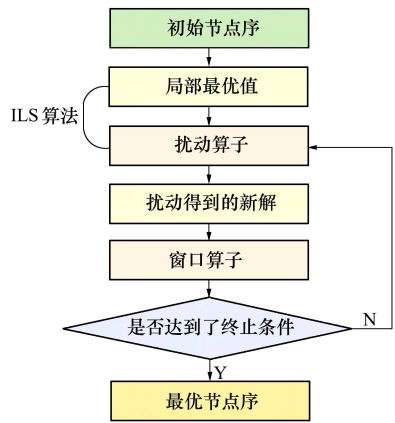


图 7 IWINOBS 算法流程图

算法 1: IWINOBS

Input: 数据集 D , 初始节点序 $prime_order$, 节点数 n , 样本量 n_s , 固定迭代次数 I_1 , 最大窗口大小 w

Output: 最优节点序 $current_order$, 最优节点序的评分 $current_score$

1. 选择初始节点序 $prime_order$ 的一个随机节点作为固定索引
2. 初始窗口大小 $w = 1$; 初始迭代次数 $I_1 = 1$;
3. While($I_1 < 4$)
4. $I_1 = I_1 + 1$;

```

5. For  $i = 1 : N - 2 * w + 1$ 
6.   While( $w \leq \text{floor}(N/2)$ )
7.      $j = i + w$ ;
8.     current_order = prime_order;
9.     temp = current_order( $i : i + w - 1$ );
10.    current_order( $i : i + w - 1$ ) = current_order( $j : j + w - 1$ );
11.    current_order( $j : j + w - 1$ ) = temp;
12.    [current_dag, current_score] = learn_struct_K2(data,  $n_s$ , current_order, 'scoring_fn', 'bic');
13.    If( $\text{current\_score} \leq \text{prime\_score}$ )
14.       $w = w + 1$ ;
15.    Else If( $\text{current\_score} > \text{prime\_score}$ )
16.      prime_order = current_order
17.      prime_score = current_score
18.    End if;
19.  End while
20. End for;
21. End while;
22. Return current_order, current_score.

```

3 实验分析

本文的实验仿真环境为 MATLAB R2018b, 操作系统为 Windows 10, CPU 为 Intel(R) Core(TM) i5-6300U CPU @ 2.40 GHz, RAM 为 8.00G。为了评价本文算法的学习性能, 基于贝叶斯网络工具箱 FullBNT-1.0.7, 使用 Asia^[17]、Car^[18] 和 Alarm^[19] 网络进行实验仿真, 3 个网络的属性说明如表 1 所示, 标准网络结构如图 8 所示。

表 1 标准贝叶斯网络的参数

数据集	节点数	有向边数	最大出度/入度	变量域
Asia	8	8	2/2	2
Car	12	9	3/3	2
Alarm	37	46	5/4	2~4

本文仿真实验对比的项目如下:

1) 算法学习效率

算法的学习效率用学习 BN 结构所需要的时间来计算。

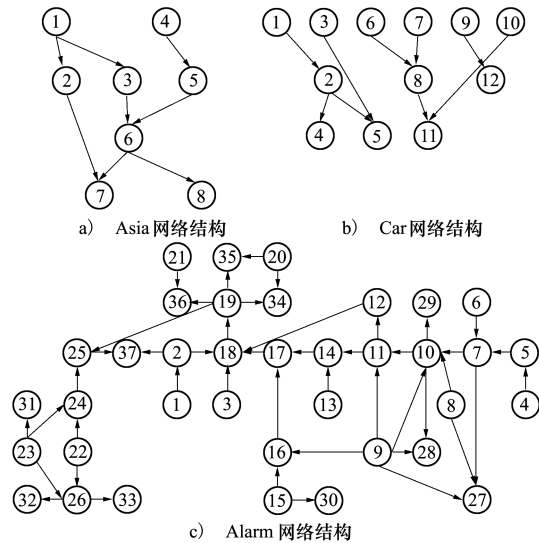


图 8 Asia、Car 和 Alarm 网络的结构图

2) 算法学习精度

①结构正确率: 算法学习出的 BN 结构的边数中正确边数所占比例。BN 结构的边数包括错误边数和正确边数, 错误边数是指与标准网络结构相比, 该算法获得的网络结构的冗余边数、反向边数和缺失边数之和; 正确边数是指与标准网络结构相比, 该算法获得的网络结构的正确边数。

②BIC 评分: 即采用该算法得到的网络的 BIC 评分, 评分值越高, 说明网络越好。

3.1 算法学习效率的比较

本实验主要比较随着节点数目的增加, OBS、IINOBS、IWINOBS 算法和网络空间下经典的 BN 结构学习算法的运行时间。利用 Asia、Car 和 Alarm 网络, 分别随机生成样本量为 1 000, 2 000, 3 000, 5 000 的数据集, 每种算法分别运行 10 次, 记录算法每次运行的时间, 取平均值。实验结果如表 2~4 所示。

表 2 Asia 网络中不同算法的运行时间对比

算法	样本量			
	1 000	2 000	3 000	5 000
HC	10.49	13.49	14.96	18.48
K2	10.36	12.74	15.68	17.37
OBS	10.10	10.79	12.24	13.15
IINOBS	9.73	11.18	13.59	16.55
IWINOBS	5.91	8.79	9.94	11.06

表 3 Car 网络中不同算法的运行时间对比 s

算法	样本量			
	1 000	2 000	3 000	5 000
HC	52.10	68.21	82.94	109.43
K2	51.80	54.42	65.30	71.76
OBS	41.30	47.97	48.79	60.25
IINOBS	38.67	45.45	53.86	66.35
IWINOBS	20.29	25.63	29.65	37.19

表 4 Alarm 网络中不同算法的运行时间对比 s

算法	样本量			
	1 000	2 000	3 000	5 000
HC	1 387.14	1 638.94	1 863.79	2 202.65
K2	246.03	306.98	487.67	511.78
OBS	131.62	167.80	202.66	272.48
IINOBS	7 006.92	8251.14	10 180.19	15 061.52
IWINOBS	9 734.94	13 016.77	16 860.37	20 937.84

通过对比表 2~4 发现,在算法的运行时间上, K2 和 HC 算法的学习时间明显高于本文的 IWINOBS 算法。原因在于 K2 算法采用遍历搜索空间来搜索最优网络结构,所以学习效率较为低下; HC 算法容易陷入局部最优,因此采用多次运行 HC 算法的方式来避免这个缺点,而爬山次数的增大也导致了算法学习效率明显降低。

在学习有 8 个节点的 Asia 网络时,本文提出的 IWINOBS 算法的学习效率与 IINOBS 算法相比提高了 30.17%,精度没有变化;与 K2 算法相比学习效率提升了 30.96%,精度仅下降了 3.41%。在学习有 12 个节点的 Car 网络时,本文算法的学习效率与 IINOBS 算法相比提高了 45.01%,精度提高了 2.33%;与 K2 算法相比学习效率提升了 54.12%,精度仅下降了 6.38%。在学习有 37 个节点的 Alarm 网络时,算法的运行时间大大增加,在学习效率方面 OBS 算法表现较好,与 K2 算法相比学习效率提升了 49.26%,学习精度仅仅降低了 3.45%。

3.2 算法学习精度的比较

3.2.1 结构正确率

贝叶斯网络结构正确率 (structural accuracy, SA) 指标一般由该算法获得的网络结构的冗余边数 (redundant edge, RE)、缺失边数 (missing edge, ME)、反向边数 (inverted edge, IE) 和正确边数 (correct edge, CE) 来衡量,其计算方法如公式 (8) 所示。

$$R_{SA} = \frac{\sum E_{CE}}{\sum E_{CE} + \sum E_{RE} + \sum E_{ME} + \sum E_{IE}} \quad (8)$$

为评价本文算法的求解质量,本实验分别对 1 000 组 Asia、Car 和 Alarm 网络进行学习,比较 OBS、IINOBS、IWINOBS 和 K2 算法学习结果的结构正确率。将每种算法分别运行 10 次,记录每次的学习结果中贝叶斯网络结构中冗余边、缺失边、反向边和正确边的情况,取平均值,算法学习结果对比如表 5~7 和图 9 所示。

表 5 Asia 网络中不同算法的 BN 结构学习结果对比

算法	RE	ME	IE	CE	SA
K2	0	1.00	0	7.00	0.88
OBS	0	1.40	0	6.60	0.83
IINOBS	0	1.20	0	6.80	0.85
IWINOBS	0.20	1.00	0	6.80	0.85

表 6 Car 网络中不同算法的 BN 结构学习结果对比

算法	RE	ME	IE	CE	SA
K2	0	0.5	0	8.50	0.94
OBS	0	1.50	0	7.50	0.83
IINOBS	0	1.30	0	7.70	0.86
IWINOBS	0.10	1.00	0	7.90	0.88

表 7 Alarm 网络中不同算法的 BN 结构学习结果对比

算法	RE	ME	IE	CE	SA
K2	1.20	4.90	0	39.90	0.87
OBS	1.50	5.70	0	38.80	0.84
IINOBS	1.70	5.50	0	38.80	0.84
IWINOBS	1.30	5.80	0	38.90	0.85

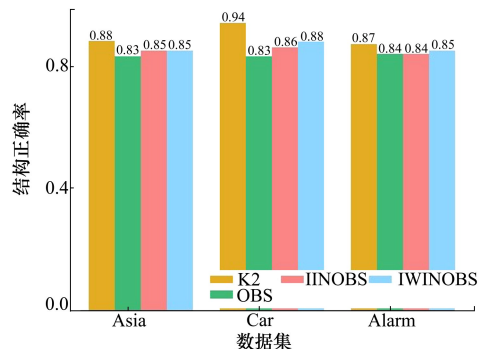


图 9 不同算法的 BN 结构正确率对比

K2 算法给定了正确的节点序作为输入,因此在学习精度方面有着明显优势。根据表 5~7 和图 9 可以得出,在学习有 8 个节点的小型 Asia 网络时,本文提出的 IWINOBS 算法学习出的 BN 结构与 K2 算法相比结构正确率仅降低 3.41%,与 IINOBS 算法相比无变化;在学习有 12 个节点的 Car 网络时,本文算法与 K2 算法相比结构正确率仅降低了 6.38%,与 IINOBS 算法相比提高了 2.33%;在学习有 37 个节点的大型 Alarm 网络时,本文算法与 K2 算法相比结构正确率仅降低了 3.45%,与 IINOBS 算法相比提高了 1.19%。通过对比第 3.1 节可知,本文算法在提高学习效率的同时能够学习出正确率较高的贝叶斯网络结构。

3.2.2 BIC 评分

本实验主要通过比较不同算法学习出的贝叶斯网络结构的 BIC 评分来评价该算法的性能。在样本量为 1 000, 2 000, 3 000, 5 000 的情况下,分别采用 K2、OBS、IINOBS、IWINOBS 算法学习 Asia、Car 和 Alarm 网络。考虑到算法的随机性,将每种算法分别运行 10 次,并记录每次结果的 BIC 评分,取平均值。实验结果如表 8 所示。

表 8 不同算法学习出 BN 结构的 BIC 评分对比

样本量	数据集	BIC 评分			
		K2	OBS	IINOBS	IWINOBS
1 000	Asia	-234.96	-219.70	-14.29	-14.19
	Car	-640.20	-544.27	-420.16	-253.22
	Alarm	-99.18	-56.80	-33.94	-44.35
2 000	Asia	-410.18	-377.62	-15.68	-15.58
	Car	-1 381.20	-1 148.05	-759.46	-538.18
	Alarm	-163.13	-156.69	-64.28	-59.49
3 000	Asia	-566.78	-554.88	-16.50	-16.39
	Car	-1 959.01	-1 844.87	-1 114.72	-1 105.45
	Alarm	-236.52	-226.22	-94.50	-91.12
5 000	Asia	-990.48	-908.64	-19.52	-17.52
	Car	-3 360.86	-2 896.58	-1 844.49	-1 790.63
	Alarm	-353.61	-346.35	-117.98	-109.82

表 8 表明,随着节点数的增加,Asia、Car 和 Alarm 网络在不同样本量下采用 4 种算法学习出 BN 结构的 BIC 评分之间的差异逐渐变大。其中最优 BIC 评分在表 8 中用加粗字体表示,可以看出,IWINOBS 算法学习出的 BIC 评分总是优于 K2 算法和 OBS 算法,在大多数情况下是优于 IINOBS 算法的。对于有 8 个节点的 Asia 网络,IWINOBS 算法在不同样本量下学习出的 BIC 评分相较于 K2 算法分别提高了 93.96%,96.20%,97.11%,98.23%,相较于 OBS 算法分别提高了 93.54%,95.87%,97.05%,98.07%;对于有 12 个节点的 Car 网络,IWINOBS 算法在不同样本量下学习出的 BIC 评分相较于 K2 算法分别提高 60.45%,61.04%,43.57%,46.72%,相较于 OBS 算法分别提高了 53.48%,53.12%,40.08%,38.18%;对于有 37 个节点的 Alarm 网络,IWINOBS 算法在不同样本量下学习出的 BIC 评分相较于 K2 算法分别提高了 55.28%,63.53%,61.47%,68.94%,相较于 OBS 算法分别提高了 21.92%,62.03%,59.72%,68.29%。由此可见,本文提出的 IWINOBS 算法与网络空间下的 K2 算法相比学习效率大大提升,但仍然可以学习到学习精度相对理想的贝叶斯网络结构。

4 结 论

本文将贝叶斯网络学习问题转化为搜索最优节点序的问题,在节点序空间中提出了一种迭代局部搜索与窗口插入算子相结合的 BN 结构学习方法——IWINOBS 算法,从而降低了算法陷入局部最优状态的概率。将本文提出的 IWINOBS 算法与目前基于节点序空间的 BN 结构学习性能较好的方法进行比较发现,本文算法能得到性能更优的网络结构。与网络空间下的经典算法如 K2、HC 算法进行比较发现,在保持算法精度损失较小的情况下,IWINOBS 算法的学习效率更高。在之后的研究工作中,将进一步优化节点序的搜索方法,在大规模网络结构学习中实现精度损失尽可能小且能够降低复杂度,进而提升大规模贝叶斯网络的学习性能。

参考文献:

- [1] 张连文, 郭海鹏. 贝叶斯网引论[M]. 北京: 科学出版社, 2006
ZHANG Lianwen, GUO Haipeng. Introduction to Bayesian nets[M]. Beijing: Science Press, 2006 (in Chinese)
- [2] CAI B, KONG X, LIU Y, et al. Application of Bayesian networks in reliability evaluation[J]. IEEE Trans on Industrial Informatics, 2018, 15(4): 2146-2157
- [3] SCUTARI M, GRAAFLAND C E, GUTIÉRREZ J M. Who learns better Bayesian network structures: constraint-based, score-based or hybrid algorithms[J]. Proceedings of Machine Learning Research, 2018, 72: 416-427
- [4] LYU Y, MIAO J, LIANG J, et al. BIC-based node order learning for improving Bayesian network structure learning[J]. Frontiers of Computer Science, 2021, 15(6): 1-14
- [5] KALTENPOTH D, VREEKEN J. We are not your real parents; telling causal from confounded using mdl[C]//Proceedings of the 2019 SIAM International Conference on Data Mining, 2019: 199-207
- [6] SCUTARI M. Dirichlet Bayesian network scores and the maximum relative entropy principle[J]. Behaviormetrika, 2018, 45(2): 337-362
- [7] SCUTARI M, VITOLO C, TUCKER A. Learning Bayesian networks from big data with greedy search: computational complexity and efficient implementation[J]. Statistics and Computing, 2019, 29(5): 1095-1108
- [8] TEYSSIER M, KOLLER D. Ordering-based search: a simple and effective algorithm for learning bayesian networks[C]//Conference on Uncertainty in Artificial Intelligence, 2005: 584-590
- [9] BEHJATI S, BEIGY H. Improved K2 algorithm for Bayesian network structure learning[J]. Engineering Applications of Artificial Intelligence, 2020, 91: 103617
- [10] WANG Y. Analysis of the max-min hill-climbing algorithm[C]//2018 International Conference on Transportation & Logistics, Information & Communication, Smart City, 2018: 509-511
- [11] LIU X, GAO X, WANG Z, et al. Improved local search with momentum for Bayesian networks structure learning[J]. Entropy, 2021, 23(6): 750
- [12] HE C, GAO X, GUO Z. Structure learning of Bayesian networks by finding the optimal ordering[C]//2018 24th International Conference on Pattern Recognition, 2018: 177-182
- [13] WANG Z, GAO X, TAN X, et al. Determining the direction of the local search in topological ordering space for Bayesian network structure learning[J]. Knowledge-Based Systems, 2021, 234: 107566
- [14] LEE C, BEEK P. Metaheuristics for score-and-search Bayesian network structure learning[C]//Canadian Conference on Artificial Intelligence, 2017: 129-141
- [15] ALONSO-BARBA J I, DELAOSSA L, PUERTA J M. Structural learning of Bayesian networks using local algorithms based on the space of orderings[J]. Soft Computing, 2011, 15(10): 1881-1895
- [16] SCANAGATTA M, CORANI G, ZAFFALON M. Improved local search in Bayesian networks structure learning[J]. Proceedings of Machine Learning Research, 2017, 73: 45-56
- [17] JIANG M, LU J. The analysis of maritime piracy occurred in Southeast Asia by using Bayesian network[J]. Transportation Research Part E: Logistics and Transportation Review, 2020, 139: 101965
- [18] Bayesian network software.[CP/OL]. (2008-03-17) [2022-06-23]. <http://www.bayesia.com>
- [19] BEINLICH I A, SUERMONDT H J, CHAVEZ R M, et al. The ALARM monitoring system: a case study with two probabilistic inference techniques for belief networks[C]//Second European Conference on Artificial Intelligence in Medicine, London, 1989

A Bayesian network structure learning method for optimizing ordering search operator

JIA Liuna, DONG Mianmian, HE Chuchao, DI Ruohai, LI Xiaoyan

(School of Electronic Information Engineering, Xi'an Technological University, Xi'an 710021, China)

Abstract: Local search algorithm in ordering space is a good method which can effectively improve the efficiency of bayesian network structure learning. However, the existing algorithms usually have problems such as insufficient order optimization, low learning accuracy, and easy stop at a local optimal. In order to solve these problems, the local search algorithm in ordering space is studied, and a new method to improve the accuracy of bayesian network structure learning by optimizing order search operator is proposed. Combining the iterative local search algorithm with the window operator to search the neighborhood of a given order in the ordering space, the probability of the algorithm falling into the local optimal value is reduced, and the network structure with higher quality is obtained. Experimental results show that comparing with the bayesian network structure learning algorithm in network structure space, the learning efficiency of the present algorithm is improved by 54.12%. Comparing with the bayesian network structure learning algorithm in ordering space, the learning accuracy of the present algorithm is improved by 2.33%.

Keywords: Bayesian network; structure learning; order optimization; search operator; local search

引用格式: 贾柳娜,董绵绵,贺楚超,等. 一种优化节点序搜索算子的 BN 结构学习方法[J]. 西北工业大学学报, 2023, 41(2): 419-427

JIA Liuna, DONG Mianmian, HE Chuchao, et al. A Bayesian network structure learning method for optimizing ordering search operator[J]. *Journal of Northwestern Polytechnical University*, 2023, 41(2): 419-427 (in Chinese)