基于节点块序列约束的局部贝叶斯网络结构搜索算法

王海羽 [1,2] 刘浩然 [1,2] 张力悦 [1,2] 张春兰 [1,2] 刘 彬 [1,2]

摘 要 针对 K2 算法过度依赖节点序和节点序搜索算法评价节点序效率较低的问题, 提出一种基于节点块序列约束的局部贝叶斯网络结构搜索算法, 该算法首先通过评分定向构建定向支撑树结构, 在此基础上构建节点块序列, 然后利用节点块序列确定每个节点的潜在父节点集, 通过搜索每个节点的父节点集构建网络结构, 最后对该结构进行非法结构修正得到最优贝叶斯网络结构. 利用标准网络将算法与几种不同类型的改进算法进行对比分析, 验证该算法的有效性.

关键词 贝叶斯网络结构学习, 定向最大支撑树, 节点块序列, K2 算法

引用格式 王海羽, 刘浩然, 张力悦, 张春兰, 刘彬. 基于节点块序列约束的局部贝叶斯网络结构搜索算法. 自动化学报, 2020, **46**(6): 1210−1219

DOI 10.16383/j.aas.c180108

Local Bayesian Network Structure Searching Using Constraint of Node Chunk Sequence

WANG Hai-Yu[1,2] LIU Hao-Ran[1,2] ZHANG Li-Yue[1,2] ZHANG Chun-Lan[1,2] LIU Bin[1,2]

Abstract The performance of the K2 algorithm depends on node ordering heavily. The Bayesian learning algorithm based on node order search needs to evaluate the quality of the node sequence by K2 algorithm, resulting in the problem of low efficiency in large and medium-sized networks. In this paper, a new Bayesian structure learning algorithm is proposed to solve the BN structure learning problem by searching local Bayesian network structure construction using node chunk sequence constraints. The algorithm firstly constructs a directional maximum weight spanning tree structure by using score orientation and generates the node chunk sequence by this structure. Then, the node chunk sequence is used to determine the potential parent node set of each node. The network structure is built by searching the parent node set of each node, and the structure is modified by illegal structure to get the optimal Bayesian network structure. Finally, some experiments are designed to evaluate the performance of the proposed algorithm, in which the standard network is used to compare the algorithm with several different improved algorithms to verify the effectiveness of the algorithm. The results indicate that the proposed algorithm is worthy of being studied in the field of BNs construction.

Key words Bayesian network structure learning, directional maximum weight spanning tree, the node chunk sequence, K2 algorithm

Citation Wang Hai-Yu, Liu Hao-Ran, Zhang Li-Yue, Zhang Chun-Lan, Liu Bin. Local Bayesian network structure searching using constraint of node chunk sequence. *Acta Automatica Sinica*, 2020, **46**(6): 1210−1219

贝叶斯网络 (Bayesian network, BN) 是表示复杂概率知识的有力工具, 通过有向无环图和条件概率表表示变量之间的联合概率分布, 有助于理解变量之间的因果关系以及数据集特征[1], 已广泛应用于医学[2]、风险评估[3]、自然语言处理[4] 等多个领域[5]. 贝叶斯网络学习主要包括结构学习和参数学习, 其中贝叶斯网络结构的准确度直接影响参数学习和推理结果精度, 但是通过数据学习贝叶斯网络结构是 NP 难的. 国内外专家学者提出了许多从数据中学习贝叶斯结构的方法, 其中最常见的是基于评分搜索的算法[6], 这类算法通过评分函数对网络结构进行评分搜索, 寻找最优得分的网络结构, 可以学习比较精确的网络结构, 但随着网络规模的增加, 搜索空间呈指数增长, 且大部分启发式评分算法在进行种群迭代时会出现大量非法结构[7], 当网络规模较大时, 对非法结构的修正会产生较高的时间复杂度, 影响算法效率.

最大支撑树算法 (Maximum weight spanning tree, MWST)[8] 通过保留互信息[9] 值最大的节点关系, 可以有效地缩减搜索空间. 然而, MWST 算法

收稿日期 2018-02-27 录用日期 2018-08-28
Manuscript received February 27, 2018; accepted August 28, 2018
国家自然科学基金 (51641609, 61802333) 资助
Supported by National Natural Science Foundation of China (51641609, 61802333)
本文责任编委 黎铭
Recommended by Associate Editor LI Ming
1. 燕山大学信息科学与工程学院 秦皇岛 066004 2. 燕山大学河北省特种光纤与光纤传感重点实验室 秦皇岛 066004
1. School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004 2. Key Laboratory for Special Fiber and Fiber Sensor of Hebei Province, Yanshan University, Qinhuangdao 066004

是通过随机选择根节点对结构中相邻的节点进行定向生成有向图, 该方法定向得到的边不具有因果语义, 结构准确度较低. Pearl[10] 利用碰撞识别判断节点间因果关系, 该算法效率较高, 但只能对具有多父节点的边进行定向; Jiang 等[11] 通过相对条件平均熵对节点间关系进行定向, 该方法操作简单但准确率较低.

K2 算法[11] 利用节点序约束搜索过程, 可以显著减少搜索空间并有效避免产生非法结构且性能优于大部分经典算法和启发式算法. 节点序对 K2 算法性能影响极大, 良好的排序需要大量的专家知识, 由于专家知识的差异性, 网络规模较大时难以实现[12]. 为此研究人员提出了许多解决方案, 如 Chen 等[13] 结合信息论, 利用条件独立测试和穷举搜索进行节点间定向并对节点进行排序, 但该算法时间复杂度较高; Ko 等[14] 提出通过条件频率的方法确定 K2 算法的节点排序, 该算法通过简化搜索策略, 降低了时间复杂度, 但数据量对算法结果影响较大; Leray 等[15] 提出 MWST-K2 (Maximum weight spanning tree-K2) 算法, 通过最大支撑树拓扑排序生成节点序, 但由于最大支撑树算法定向的精度较低, 导致运行结果并不理想; Faulkner[16] 提出 K2GA (K2-genetic algorithm) 算法, 以节点序视为遗传算法的染色体, 通过修改后的 K2 算法对每条染色体进行评分得到种群适应度, 并进行种群迭代, 将贝叶斯网络结构学习问题转化为节点序搜索问题; Wu 等[17] 利用蚁群算法对 K2GA 算法中的遗传算法进行替换, 提出 K2ACO (K2-ant colony optimization) 算法; Aouay 等[18] 提出 PSOK2 (Particle swarm optimization-K2) 算法, 通过粒子群算法对节点序进行更新, 利用节点序作为 K2 算法的先验知识进行贝叶斯网络构建, 通过 AIC (Akaike information criterion) 评分函数对网络结构进行评分得到对应节点序的适应度函数进行种群迭代, 但算法效率较低, 运行速度较慢; 刘浩然等[19] 提出 MAK (MWST-ACO-K2 (Maximum weight spanning tree-ant colony optimization-K2)) 算法, 该算法融合最大支撑树和蚁群算法搜索节点序, 在处理小型网络时可取得较理想的结果, 与 K2GA、K2ACO、PSOK2 等基于节点序搜索的贝叶斯结构学习算法类似, 需要对种群中所有个体运行 K2 算法得到对应的适应度值, 才能对种群进行更新迭代, 存在算法时间复杂度较高, 不同的参数设置对结果影响较大, 在大中型网络效果不理想等问题.

本文提出一种基于节点块序列约束的局部贝叶斯网络结构搜索算法 (Bayesian network construction based on node chunk sequence constraints) — NCSC 算法. 算法首先对最大支撑树进行评分定向构建定向支撑树结构, 利用该结构搜索得到节点块序列. 之后利用节点块序列作为搜索顺序, 将定向支撑树结构作为先验知识, 通过节点块序列确定每个节点的潜在父节点集, 对每个节点的父节点集进行搜索, 构建初始网络结构. 最后对该结构进行非法结构修正, 得到最优贝叶斯网络结构. 通过将位于相同节点块的节点视为等价节点, 避免对节点块内元素进行排列, 解决了基于节点序搜索的贝叶斯结构学习算法评价节点序时间代价严重的问题, 可以在不借助专家知识的情况下得到较为精确的贝叶斯网络结构.

1 NCSC 算法研究

1.1 NCSC 算法构建

设 $B=(G,\Theta)$ 是一个以集合 X 为节点的贝叶斯网络结构[20], 其中 $X=\{X_1,X_2,\cdots,X_n\}$, X_i 的取值范围为 $\{x_{i1},x_{i2},\cdots,x_{ir_i}\}$, $G=(V,E)$ 为有向无环图, $V=\{X_1,X_2,\cdots,X_n\}$ 表示节点集, 有向边集合 E 表示节点之间的依赖关系, Θ 为一组表示每个节点 X_i 条件概率分布 $P(X_i|Pa(X_i))$ 的参数, 表示 G 中节点 X_i 的父节点. 根据文献 [9], 可利用式 (1) 计算任意两节点之间的互信息.

$$I(X_i,X_j)=\sum_{i=1}^{r_i}\sum_{j=1}^{r_j}P(X_i=x_i,X_j=x_j)\times \lg\frac{P(X_i=x_i,X_j=x_j)}{P(X_i=x_i),P(X_j=x_j)} \tag{1}$$

$P(X_i=x_i,X_j=x_j)$ 表示随机变量 X_i 和 X_j 的联合概率, $P(X_i=x_i)$ 表示随机变量 X_i 的样本值为 x_i 的概率, r_i 表示离散状态的数量. 从变量集合 X 中选择任意节点 X_i 作为起始点, 设集合 $U=\{X_i\}$.

从集合 $X-U$ 中搜索节点 X_j, 使其满足节点 X_j 到集合 U 中任意一个节点的互信息值 $I(X_j,u)$ $(u\in U)$ 最大, 在不产生环路的情况下通过无向边连接 X_j 和节点 u, 并将节点 X_j 加入集合 U; 以此类推, 当集合 U 包含全部节点时, 可得到节点间互信息值之和 $W(t)$ 为最大值时所对应最大支撑树结构, $W(t)$ 如式 (2) 所示.

$$W(t)=\sum_{X_i\in U,X_j\in X-U}I(X_i;X_j) \tag{2}$$

$$I(X,Y)=I(Y,X) \tag{3}$$

由式 (3) 可知, 互信息具有对称性, 即此时的最大支撑树结构中仅存在双向边. 利用空网络初始

化定向支撑树 T, 利用式 (4) 和式 (5), 依次将最大支撑树结构中的双向边 $E(X \leftrightarrow Y)$ 分别以单向边 $E(X \rightarrow Y)$ 和 $E(X \leftarrow Y)$ 的形式添加至当前定向支撑树网络结构 T, 得到子结构 $T1$ 和 $T2$.

$$T1 = T + E(X \rightarrow Y) \tag{4}$$

$$T2 = T + E(X \leftarrow Y) \tag{5}$$

将网络结构代入式 (6), 分别计算当前网络结构 T 的 CH (Cooper-Herskovits) 得分[10] $S(T, D)$ 以及添加有向边后形成的子网络结构 $T1$, $T2$ 的得分 $S(T1, D)$, $S(T2, D)$.

$$S\left(G, D\right) = \sum_{i=1}^{n} Score(X_i, Pa(X_i)) \tag{6}$$

$$Score(X_i, Pa(X_i)) = \sum_{j=1}^{q_i}\left(\lg\frac{(r_i-1)!}{(N_{ij}+r_i-1)!} + \sum_{k=1}^{r_i}\lg\left(N_{ijk}!\right)\right) \tag{7}$$

$$N_{ij} = \sum_{k\,=\,1}^{r_i} N_{ijk} \tag{8}$$

其中, G 为网络结构, D 为数据样本, n 为随机变量 (结构中的节点) 个数, r_i 为变量 X_i 的状态数, q_i 表示 X_i 的父节点 $Pa(X_i)$ 可能取值的状态数, N_{ijk} 表示数据集中, $X_i = x_k$ 且 $Pa(X_i)$ 为状态 j 的样本数量. 将得到的评分值 $S(T, D)$, $S(T1, D)$ 和 $S(T2, D)$ 代入式 (9), 通过保留双向边中能增加当前网络结构评分的单向边, 更新定向支撑树结构. 遍历最大支撑树结构, 依次对结构中的所有双向边进行定向操作, 直到最大支撑树结构中的双向边全部转化为单向边, 得到定向支撑树结构 $Tree$.

$$Tree = \begin{cases} T1, & S(T1, D) > S(T2, D) \\ & S(T1, D) > S(T, D) \\ T2, & S(T2, D) > S(T1, D) \\ & S(T2, D) > S(T, D) \\ T, & \text{其他} \end{cases} \tag{9}$$

将定向支撑树记为 $Tree(t)$, $t = 0, 1, \cdots, Ts$, 利用式 (10) 计算 $Tree(t)$ 中每个节点 X_i 的父节点数 PA_t. 将 PA_t 代入式 (11) 搜索 $Tree(t)$ 中没有父节点的节点集.

$$PA_t\left(d_i\right) = \text{num}\left(Pa\left(X_i\right)\right), \ \ X_i \in Tree\left(t\right) \tag{10}$$

$$Ind_t\left(X_{Ii}\right) = X_{Pi}, \ \ PA_t\left(X_{Pi}\right) = 0 \tag{11}$$

$$Order\left(t\right) = \{Ind_t\} \tag{12}$$

将 Ind_t 代入式 (12) 构建节点 $Order(t)$, 将位于相同节点块内的不同节点定义为等价节点, 之后对 $Tree(t)$ 进行更新, 删除 $Tree(t)$ 中集合 Ind_t 所含节点, 并删除与这些节点相连的有向边, 将更新后的定向支撑树结构用 $Tree(t+1)$ 表示, 利用式 (10) ∼ (12) 得到 $Tree(t+1)$ 所对应的 PA_{t+1}, Ind_{t+1}, 以及节点块 $Order(t+1)$. 以此类推, 当更新后的定向支撑树中不包含任何节点时停止搜索, 将节点块代入式 (13) 得到节点块序列 $\boldsymbol{Ord}$, 且节点块序列唯一[21]. 节点块序列搜索过程示意图如图 1 所示, 其中虚线表示被删除的节点和有向边, $Tree(0)$ 为定向支撑树.

$$\boldsymbol{Ord} = \left[Order\left(1\right), Order\left(2\right), \cdots, Order\left(Ts\right)\right] = \left[X_{O1}, X_{O2}, \cdots, X_{On}\right] \tag{13}$$

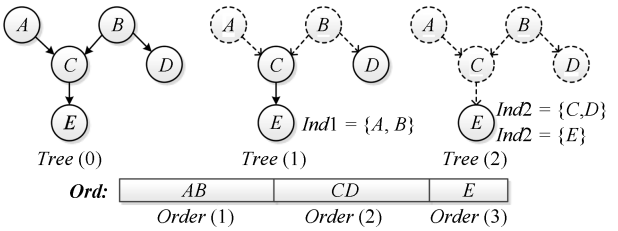

图 1 由定向支撑树构建节点块序列的例子

Fig. 1 Example of constructing node chunk sequence by directional support tree

将定向支撑树作为搜索算法的先验信息, 利用式 (14) 分别计算节点 X_i 的父节点集为空集时的贝叶斯信息准则 (Bayesian information criterion, BIC)[22] 评分 $S_{\text{BIC}}((X_i, \emptyset)\,|D)$ 和节点 X_i 的父节点集为定向支撑树 $Tree$ 中 X_i 的父节点集时的 BIC 评分 $S_{\text{BIC}}((X_i, Pa_T(X_i))\,|D)$.

$$S_{\text{BIC}}((X_i, Pa(X_i))\,|D) = \sum_{j=1}^{q_i}\sum_{k=1}^{r_i}\left(N_{ijk}\lg\frac{N_{ijk}}{N_{ij}} - \frac{1}{2}q_i\left(r_i-1\right)\lg N\right) \tag{14}$$

其中, N 表示样本数量, 式 (14) 中负项表示网络结构复杂度的惩罚项. 将评分代入式 (15) 确定每个节点的初始潜在父节点集 $Pa(X_i)$, 并记其 BIC 得分为 S_{Old}, 如式 (16) 所示.

$$Pa(X_i) = \begin{cases} \emptyset, & S_{\text{BIC}}((X_i, \emptyset)\,|D) \geq \\ & S_{\text{BIC}}((X_i, Pa_T(X_i))\,|D) \\ Pa_T(X_i), & S_{\text{BIC}}((X_i, \emptyset)\,|D) < \\ & S_{\text{BIC}}((X_i, Pa_T(X_i))\,|D) \end{cases} \tag{15}$$

$$S_{\mathrm{Old}} = S_{\mathrm{BIC}}((X_i, Pa(X_i))\,|D) \tag{16}$$

将节点块序列作为搜索顺序, 利用式 (17) 依次选取节点块序列中的变量 X_{Oi} 作为当前搜索节点 X_i, 将节点块序列中, 排在 X_i 之前且与 X_{Oi} 位于相同节点块 $Order(t_i)$ 内, 将不属于 $Pa(X_i)$ 的节点作为 X_i 的潜在父节点集 P. 假设图 2 中节点 C 为当前搜索节点, 则 C 的潜在父节点集 P 为 $Order(1) + Order(2) - C$, 即 ABD.

$$P = \{Order\,(1)\,, Order\,(2)\,, \cdots, Order\,(t_i)\} - \{X_i, Pa\,(X_i)\} \tag{17}$$

图 2　由节点块序列搜索潜在父节点集的简单例子

Fig. 2　A simple example of searching potential parent set by node chunk sequence

从 P 中选择节点 X_j 加入 $Pa(X_i)$, 利用式 (18) 计算当前结构评分 S_{New}, 利用式 (19) 在 P 中搜索能够最大化当前结构得分的节点 X_j.

$$S_{\mathrm{New}} = S_{\mathrm{BIC}}((X_i, Pa(X_i) \cup X_j)\,|D) \tag{18}$$

$$X_j = \arg \max_{X_j \in \{P\}} S_{\mathrm{BIC}}((X_i, Pa(X_i) \cup X_j)\,|D) \tag{19}$$

利用式 (20) 更新 X_i 的父节点集 $Pa(X_i)$, 将结果代入式 (16), 更新 S_{Old}. 重复操作, 当潜在父节点集中不存在能够提升当前结构评分的节点时, 完成对节点 X_i 的父节点集 $Pa(X_i)$ 的搜索. 以节点块序列为搜索顺序, 对所有节点的父节点集进行搜索. 将搜索结果代入式 (21), 根据网络结构的可分解性得到初始网络结构 G.

$$Pa(X_i) = \begin{cases} Pa(X_i) + X_j, & S_{\mathrm{New}} > S_{\mathrm{Old}} \\ Pa(X_i), & S_{\mathrm{New}} \le S_{\mathrm{Old}} \end{cases} \tag{20}$$

$$G = \sum_{i=1}^{n} (X_i, Pa\,(X_i)) \tag{21}$$

为保证 G 为有向无环图, 需要对网络进行环路和双向边的判断及修正. 通过删除 G 中没有父节点的节点, 没有子节点的节点以及与它们相连接的有向边, 对网络 G 进行更新. 当更新后的网络 G_c 中不存在该类节点时, 停止对网络的删除操作, 若此时 G_c 中不包含任何节点, 则 G 为有向无环图; 否则, G_c 即为 G 中存在的非法结构, 需对该结构进行修正.

将 G_c 中的双向边集合记为 E_F, 删除 G_c 中的双向边得到网络结构 G'. 若 G' 中存在环路, 利用式 (22) 对 G' 中环路结构的每条有向边 $X_j \to X_i$ 进行评分, 将得分代入式 (23), 通过删除得分最小的有向边对 G' 进行去环操作, 得到有向无环图 G_{DAG}.

$$Score_{\mathrm{BIC}}\,(G, D) = \sum_{i=1}^{n} S_{\mathrm{BIC}}(X_i, Pa(X_i)) \tag{22}$$

$$G_{\mathrm{DAG}} = G' - \arg\min Score_{\mathrm{BIC}}((X_i, X_j)\,|D) \tag{23}$$

将当前有向无环图 G_{DAG} 记为 G_{DAG0}, 将集合 E_F 中的双向边 $E_{Fi}(X \leftrightarrow Y)$ 分别以单向边 $E_{Fi}(X \to Y)$ 和 $E_{Fi}(X \leftarrow Y)$ 的形式添加到 G_{DAG0} 中, 得到子结构 G_{DAG1} 和子结构 G_{DAG2}, 将子结构中合法结构记为集合 G_{DAGA} , 利用式 (6) 计算每个结构的评分, 将得分代入式 (24) 更新 G_{DAG}. 最终得到贝叶斯网络结构 G_{DAG}.

$$G_{\mathrm{DAG}} = \arg\max S\,(\{G_{\mathrm{DAG0}}, G_{\mathrm{DAGA}}\}\,|D) \tag{24}$$

1.2　NCSC 算法实现

NCSC 算法伪代码如下:

1) 根据式 (2) 计算节点间互信息, 构建 MWST;
2) **for** all $E \in$ MWST **do**　//E 为 MWST 中双向边
3)　将 E 中双向边利用式 (4) 和式 (5) 构建 $T1$, $T2$;
4)　利用式 (9) 构建定向支撑树 $Tree$;
5) **end for**
6) 利用 $Tree$ 和式 (13) 构建节点块序列 $\boldsymbol{Ord}$;
7) **for** $i = Ord(1)$ to $Ord(n)$ **do**
8)　利用式 (15) 计算 $Pa(X_i)$, 利用式 (16) 计算 S_{Old};
9)　OK_to_proceed = true;
10)　**while** OK_to_proceed **do**
11)　利用式 (20) 在 P 中选择节点 X_j;
12)　**if** $S_{\mathrm{Old}} < S_{\mathrm{New}}$ **then**
13)　$S_{\mathrm{Old}} = S_{\mathrm{New}}$;
14)　$Pa(X_i) = Pa(X_i) \cup \{X_j\}$;
15)　**else** OK_to_proceed = false;
16)　**end if**
17)　**end while**
18)　$G = G + (X_i, Pa(X_i))$;
19) **end for**
20) **if** G 存在非法结构
21)　利用式 (23) 修正 G 中的环路;
22)　利用式 (24) 修正 G 中的双向边;
23) **end if**
24) 输出贝叶斯网络结构 G_{DAG}.

2　NCSC 算法时间复杂度分析

假设 n 为网络节点个数, m 为样本的数量, 计算每个节点间互信息的时间复杂度为 $\mathrm{O}(m)$, 需计算 $n(n-1)/2$ 次, 则计算整体互信息时间复杂度为

$O(mn^2)$, 利用互信息构建最大支撑树为 $O(n^2)$, 对支撑树定向为 $O(2mn)$, 因此, 构建定向支撑树的时间复杂度为 $O((m+1)n^2+2mn)$; 在定向支撑树的基础上搜索节点块序列的时间复杂度为 $O(n+E)$, 其中 E 为定向支撑树中有向边的数量, 具体值为 $n-1$, 利用节点块序列搜索网络结构的时间复杂度为 $O(mrk^2n^2)$, 其中 k 为父节点个数, r 表示每个节点的取值个数; 对更新后网络结构进行环路检查的时间复杂度为 $O(n+EG)$, 对非法结构进行修正的时间复杂度为 $O(m(2ED+EC))$, 其中 EG 为更新后网络结构的有向边数, ED 表示结构中双向边数量, EC 表示环路的边数, 且 EG、ED 和 EC 均小于 n, 因此修复非法结构的时间复杂度最大为 $O((3m+2)n)$; NCSC 算法时间复杂度为 $O((rk^2+m+1)n^2)$.

3 NCSC 算法仿真实验

为了验证 NCSC 算法性能, 在 MATLAB 2014a 环境中基于贝叶斯网络工具箱 FullBNT-1.0.7[23] 使用标准 Alarm 网络、Insurance 网络、Hailfinder 网络进行仿真实验, 网络的属性说明如表 1 所示.

表 1 标准贝叶斯网络的参数

Table 1 Parameters of standard Bayesian networks

网络	节点数	边数	变量域	最大节点出入度
Alarm	37	46	2～4	6
Insurance	27	52	2～5	9
Hailfinder	56	66	2～11	17

为检验算法对样本集数量的敏感程度, 利用每个网络分别经过逻辑抽样产生样本容量为 1 000, 2 000, 3 000, 5 000 的模拟数据作为实验数据. 为降低数据随机性的影响, 分别对三个网络的每种样本容量产生 10 组样本数据, 将算法在每组数据样本独立运行 10 次, 即对每个网络的不同样本容量分别运行算法 100 次, 结果取平均值. 仿真实验运行环境为: 处理器 Intel(R) Core™ i3 3240 CPU, 主频 3.4 GHz, 内存 4 GB, Windows 7 32 bit 操作系统. 为评价算法性能, 利用评价贝叶斯网络结构学习算法的常用度量[24] 对本文算法和对比算法进行分析比较, 比较的项目有:

准确边数 (C): 与标准网络相比, 算法得到的网络的正确边的数量.

冗余边数 (A): 在算法得到的网络中存在, 但不存在于标准网络中的边的数量.

反向边数 (R): 同时存在于标准网络和算法得到的网络, 但方向相反的边的数量.

缺失边数 (M): 与标准网络相比, 算法得到的结构没有学习到的边的数量.

错误边数 (W): 算法所得结构与标准网络结构差异的边数之和, 等于 A + R + M.

BIC 评分 (BIC): 算法得到的网络的 BIC 评分, 分数越高, 网络越好.

运行时间 (Ext): 算法学习贝叶斯网络所需时间.

每个网络的标准结构的平均 BIC 得分如表 2 所示.

表 2 标准贝叶斯网络结构平均 BIC 得分

Table 2 Average BIC score in standard Bayesian network structure

网络	1 000	2 000	3 000	5 000
Alarm	−10 874.35	−20 382.51	−30 057.28	−48 056.66
Insurance	−15 998.20	−30 103.21	−43 863.15	−73 069.35
Hailfinder	−67 046.62	−126 837.31	−176 348.73	−279 766.12

3.1 与标准节点序的 K2 算法对比

本节将 NCSC 算法与以标准节点序作为先验知识的 K2 算法进行仿真对比, 分别在 Alarm 网络、Insurance 网络、Hailfinder 网络上运行两种算法, 仿真结果如表 3～5 所示.

由表 3～5 中的数据可知, NCSC 算法在不需要由专家知识提供节点的拓扑序的条件下, 运行结果的质量接近以标准节点序作为先验节点序的 K2 算法, 降低了算法对先验知识的依赖. 对比表 2 中数据可知, NCSC 算法对数据量并不敏感, 即使在训练数据较小 (1 000 组数据) 的情况下也不易发生过拟合现象.

3.2 与其他算法对比

NCSC 算法本质上是对 MWST 算法进行改进的节点序搜索算法, 为分析 NCSC 算法性能, 将算法与不同类型的性能较好的贝叶斯网络结构学习算法进行仿真对比, 包括基于节点序搜索算法: MWST-K2 算法 (K2 + T)[15] 和 MWST-ACO-K2 算法 (MAK)[19]; 基于评分搜索算法: 简化爬山法 (Simplify hill-climbing, SHC)[25], 以及混合搜索算法: SAR 算法 (Separation and reunion)[26]. 其中 K2 + T 算法是较为经典的利用 MWST 进行节点序搜索算法, 并且 K2 + T 和 MAK 与 NSCS 类似, 都是利用 MWST 算法压缩节点序搜索空间, SHC 是利用 MWST 压缩评分搜索空间的算法, SAR 是近几年混合算法中性能较为良好的贝叶斯结构学习算法.

在 Alarm 网络、Insurance 网络、Hailfinder 网

表 3　NCSC 算法与标准节点序的 K2 算法在 Alarm 网络中运行结果

Table 3　Results of NCSC algorithm and standard node sequence K2 algorithm in Alarm network

Alarm		BIC	Ext (s)	结构				
				C	M	R	A	W
1 000	NCSC	−11 410.83	32.98±1.82	41.8	2.1	2.8	6.5	11.4
	K2	−11 189.51	10.27±0.61	44.2	1.8	0	4.3	6.1
2 000	NCSC	−20 791.86	42.14±1.88	41.7	1.9	2.2	6.2	10.3
	K2	−20 605.99	11.53±0.47	44.5	1.5	0	2.7	4.2
3 000	NCSC	−31 025.28	45.99±5.93	42.9	1.3	2.2	4.2	7.7
	K2	−30 505.01	14.04±1.95	44.9	1.1	0	3.1	4.2
5 000	NCSC	−49 812.61	56.25±2.13	43.2	1.2	1.1	4.7	7
	K2	−49 235.77	16.51±0.74	45.1	1.0	0.0	3.1	4.1

表 4　NCSC 算法与标准节点序的 K2 算法在 Insurance 网络中运行结果

Table 4　Results of NCSC algorithm and standard node sequence K2 algorithm in Insurance network

Insurance		BIC	Ext (s)	结构				
				C	M	R	A	W
1 000	NCSC	−16 486.31	17.28±0.34	38.3	11.9	2.3	2.6	16.8
	K2	−16 219.71	4.58±0.25	39.7	12.3	0	0.3	12.6
2 000	NCSC	−30 756.3	19.61±1.13	39.1	10.7	2.4	3.2	16.3
	K2	−30 544.86	5.11±0.31	42.2	9.8	0	0.5	10.3
3 000	NCSC	−45 369.23	24.34±3.22	39.8	10.2	2.3	3.3	15.8
	K2	−44 748.8	6.32±0.40	42.6	9.4	0	0.5	9.9
5 000	NCSC	−73 566.77	29.92±2.91	40.6	9.4	2.3	3.2	14.9
	K2	−73 148.28	7.68±0.45	43.8	8.2	0	0.6	8.8

表 5　NCSC 算法与标准节点序的 K2 算法在 Hailfinder 网络中运行结果

Table 5　Results of NCSC algorithm and standard node sequence K2 algorithm in Hailfinder network

Insurance		BIC	Ext (s)	结构				
				C	M	R	A	W
1 000	NCSC	−78 396.62	232.18±16.31	47.8	16.3	2.8	9.7	29.3
	K2	−78 157.99	25.71±1.55	48.3	17.7	0	8.1	25.8
2 000	NCSC	−131 604.2	250.58±7.94	50.2	13.9	1.6	10.1	25.6
	K2	−130 945.5	30.22±1.51	51.6	14.4	0	9.1	23.5
3 000	NCSC	−182 167.2	292.22±20.01	51.5	13.4	1.4	10.3	25.1
	K2	−181 831.9	36.58±1.84	52.9	13.1	0	9.5	22.6
5 000	NCSC	−283 288.4	356.14±16.77	51.8	12.8	1.8	10.1	24.7
	K2	−282 937.2	43.37±1.83	53.5	12.5	0	9.9	22.4

络中分别运行 NCSC 算法和上述对比算法. 根据文献 [15], 将节点编号最小的节点作为 MWST-K2 算法的根节点. 根据文献 [18], 设 MAK 中信息素强度系数为 1, 信息素蒸发系数为 0.7, 启发式因子权重为 2, 群规模为 50. 根据文献 [25] 将 SHC 算法根据节点编号由小到大进行初步定向. 根据文献 [26] 分别将 SAR 中阈值 q, λ_q 和 λ_{ql} 设置为 1, 0.8 和 0.85. 记录仿真对比结果正确边数、错误边数如图 3～5 所示, 算法的平均运行时间及标准差对比如表 6～8 所示.

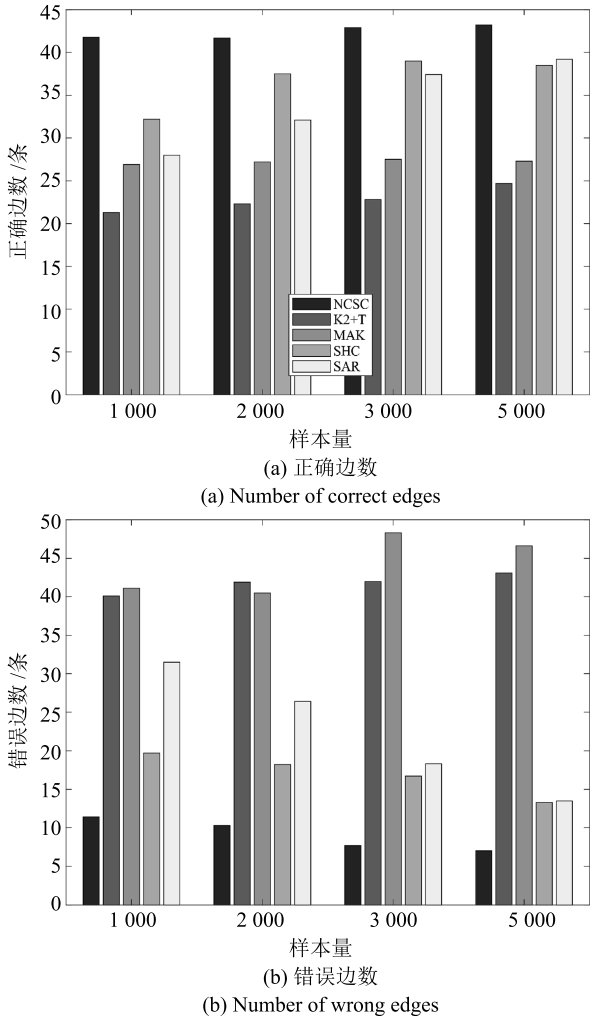

图 3 Alarm 网络中不同算法精度对比

Fig. 3 Comparison of different algorithm accuracy in Alarm network

对比图 3～5 可知, NCSC 算法可以搜索到准确度相对较高的贝叶斯网络, 算法的正确边数随着样本容量的增多而增加, 错误边数随之降低. 虽然在 Insurance 网络样本容量为 1 000 和 5 000 时准确度稍低于 SHC 算法, 但在其他仿真实验中, NCSC 算法的平均正确边数、平均错误边数均优于对比算法. NCSC 算法通过对 MWST 进行评分定向得到较为

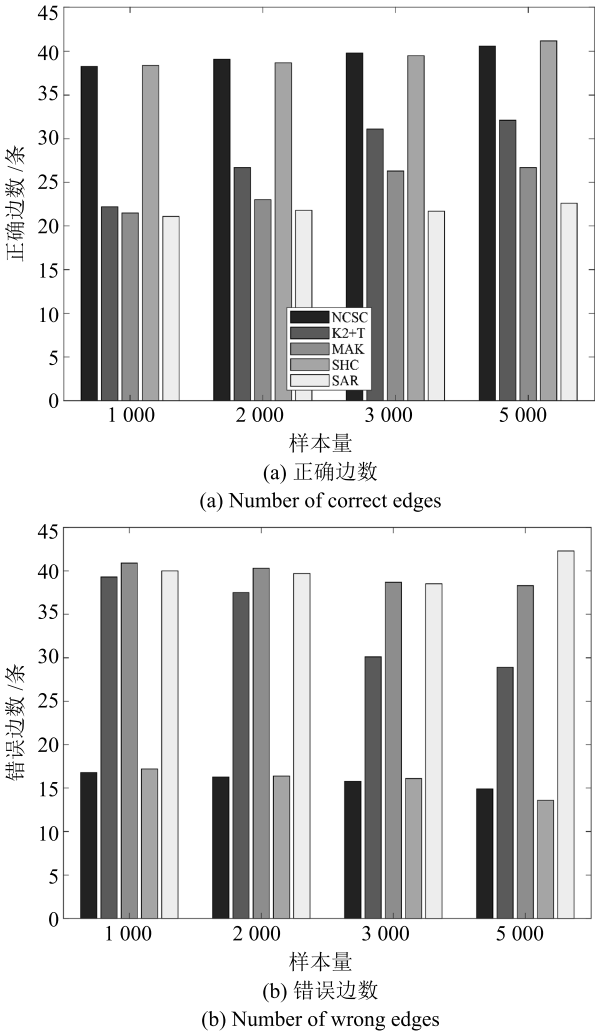

图 4 Insurance 网络中不同算法精度对比

Fig. 4 Comparison of different algorithm accuracy in Insurance network

精确的定向支撑树结构, 利用该结构进行拓扑排序得到节点块序列, 搜索每个节点 X 的父节点集时, 将位于相同节点块的节点视为等价节点, 通过将 X 之前的节点和与节点 X 位于相同节点块的等价节点作为潜在父节点, 避免对节点块内元素进行排序, 在一定程度上弥补了生成的节点块序列与标准节点序之间的误差, 提升了算法的准确度. 与 MWST-K2 算法、MAK 算法、SHC 算法、SAR 算法相比, 平均正确边数在 Alarm 网络中分别提升 86.5 %, 54.3 %, 15.8 % 和 14.91 %; 在 Insurance 网络中分别提升 43.3 %, 62.8 %, 0.01 % 和 3.01 %, 在 Hailfinder 网络中分别提升 84.6 %, 43.9 %, 6.6 %, 以及 41.14 %. 平均错误边数在 Alarm 网络中分别降低 52.4 %、降低 59.6 %、升高 1.1 %、降低 42.14 %; 在 Insurance 网络中分别降低 78.1 %, 78.9 %, 46.7 % 和 21.58 %; 在 Hailfinder 网络中分别降低 57 %, 50.2 %, 6.4 % 和 1.96 %.

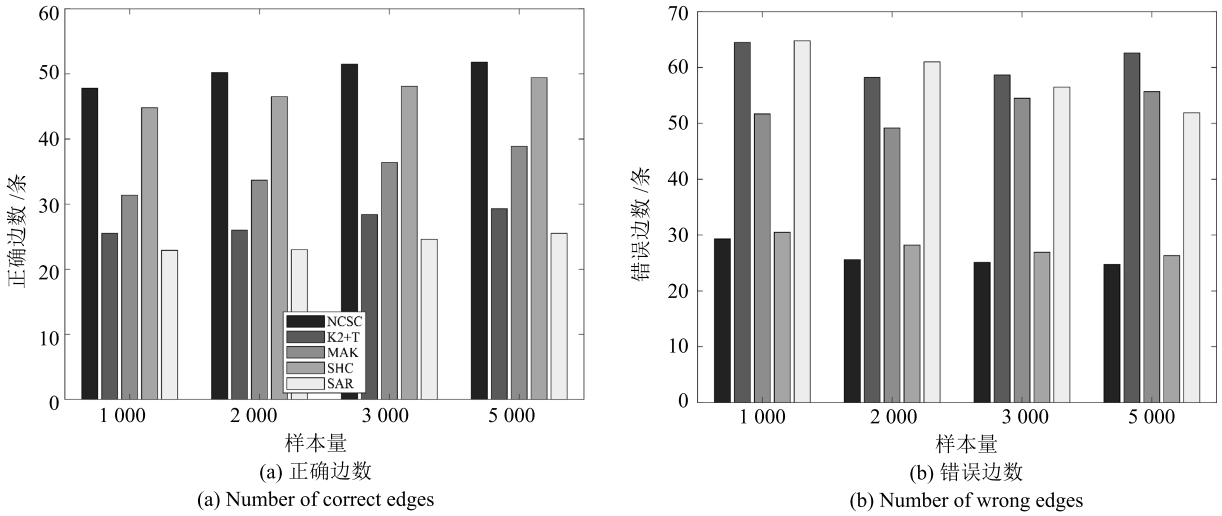

图 5　Hailfinder 网络中不同算法精度对比

Fig. 5　Comparison of different algorithm accuracy in Hailfinder network

表 6　五种算法在 Alarm 网络中运行时间 (s)

Table 6　Running time of five algorithms in Alarm network (s)

Alarm	1 000	2 000	3 000	5 000
NCSC	32.98±1.82	42.14±1.88	45.99±2.04	56.25±2.13
K2+T	11.72±0.74	14.82±1.62	16.33±1.25	18.82±2.71
MAK	942.60±31.4	1 153.37±32.37	1 299.01±42.55	1 582.35±46.86
SHC	3 474.01±90.80	4 079.66±121.98	4 784.10±156.85	6 013.93±216.64
SAR	46.77±0.81	49.64±3.52	62.30±3.38	77.39±10.91

表 7　五种算法在 Insurance 网络中运行时间 (s)

Table 7　Running time of five algorithms in Insurance network (s)

Insurance	1 000	2 000	3 000	5 000
NCSC	17.28±0.34	19.61±1.13	24.34±3.22	29.92±2.91
K2+T	4.55±0.41	6.03±1.20	7.17±0.83	9.42±2.59
MAK	388.99±12.41	420.61±27.73	481.50±32.86	694.92±30.30
SHC	4 051.46±123.82	5 521.37±179.04	5 701.31±207.10	7 072.01±241.25
SAR	19.38±0.90	19.95±1.55	31.67±4.81	40.60±5.95

表 8　五种算法在 Hailfinder 网络中运行时间 (s)

Table 8　Running time of five algorithms in Hailfinder network (s)

Hailfinder	1 000	2 000	3 000	5 000
NCSC	232.18±6.31	250.58±7.94	292.22±10.73	356.14±14.86
K2+T	28.54±0.92	32.82±0.93	39.21±1.16	49.38±1.76
MAK	2 203.76±61.58	2 611.30±66.94	3 100.82±96.87	3 854.99±120.45
SHC	18 273.69±468.55	23 500.71±318.11	29 255.29±380.52	35 785.38±420.00
SAR	229.07±7.82	264.35±15.53	307.52±12.35	362.27±20.55

由表 6～8 可知, NCSC 算法运行时间高于 MWST-K2 算法且与 SAR 算法的时间消耗处于相同数量级, 但由图 3～5 可知, NCSC 算法的准确度在三个网络中的运行结果均高于 MWST-K2 算法, 原因在于 NCSC 算法利用评分搜索对最大支撑树定向, 而 MWST-K2 算法通过随机选取一个节点作为根节点, 以根节点作为父节点对结构中相邻的节点进行定向构建定向支撑树, 这种方法虽然耗时较少, 但定向结果不具有因果语义且精度较低; 与 MAK 算法、SHC 算法相比, NCSC 算法运行速度明显提升, 在 Alarm 网络、Insurance 网络、Hailfinder 网络中, MAK 算法的平均运行时间分别是 NCSC 算法的 28.1 倍、21.7 倍和 10.4 倍; SHC 算法运行时间分别是 NCSC 算法的 103.6 倍、245.1 倍和 94.4 倍. 虽然 SHC 算法通过在初步定向的最大支撑树上运行进行爬山法可搜索到精确度较好的贝叶斯网络结构, 但是运行该算法所需的时间代价过高, 在大中型网络中难以接受. NCSC 算法通过定向支撑树以及节点块序列搜索缩减搜索空间, 与 MAK 算法相比, NCSC 算法利用节点块序列代替节点序限制搜索空间, 避免对每条节点序代入 K2 算法进行评分迭代, 降低了搜索时间. 与 SHC 算法相比, NCSC 算法在搜索网络结构时利用评分函数的可分解性, 对每个节点及其父节点集进行局部搜索评分, 时间复杂度低于 SHC 算法采用的整体贝叶斯网络结构评分.

4 结束语

针对 K2 算法过度依赖节点序, 节点序搜索算法评价节点序质量时间代价严重的问题. 提出了一种基于节点块序列约束的局部网络结构搜索算法, 通过构建定向支撑树生成节点块序列, 同时将相同节点块内的节点视为等价节点, 有效地确定了每个节点的潜在父节点集, 扩展了每个节点集的搜索范围, 避免对节点块内元素进行排列, 解决了基于节点序搜索的贝叶斯结构学习算法评价节点序时间代价严重的问题. 利用每个节点的父节点集构建初始网络降低了非法结构修正的次数. 仿真实验表明, 在不使用节点序作为先验知识且无需人为设置复杂参数的情况下, 算法运行结果接近以标准节点序作为先验节点序的 K2 算法的精度, 与不同种类的改进算法相比, NCSC 算法在精度和运行速度上都有不同程度的提高, 相比于简化爬山法运行速度平均提升了 10^2 倍以上.

References

1 Tien I, Der Kiureghian A. Algorithms for Bayesian network modeling and reliability assessment of infrastructure systems. *Reliability Engineering and System Safety*, 2016, **156**: 134−147

2 Gendelman R, Xing H M, Mirzoeva O K, Sarde P, Curtis C, Feiler H S, et al. Bayesian network inference modeling identifies TRIB1 as a novel regulator of cell-cycle progression and survival in cancer cells. *Cancer Research*, 2017, **77**(7): 1575−1585

3 Lee D, Pan R. A nonparametric Bayesian network approach to assessing system reliability at early design stages. *Reliability Engineering and System Safety*, 2018, **171**: 57−66

4 Chaturvedi I, Ragusa E, Gastaldo P, Zunino R, Cambria E. Bayesian network based extreme learning machine for subjectivity detection. *Journal of the Franklin Institute*, 2018, **355**(4): 1780−1797

5 Khakzad N, Van Gelder P. Vulnerability of industrial plants to flood-induced natechs: a Bayesian network approach. *Reliability Engineering and System Safety*, 2018, **169**: 403−411

6 Liu Jian-Wei, Li Hai-En, Luo Xiong-Lin. Learning technique of probabilistic graphical models: a review. *Acta Automatica Sinica*, 2014, **40**(6): 1025−1044
(刘建伟, 黎海恩, 罗雄麟. 概率图模型学习技术研究进展. 自动化学报, 2014, **40**(6): 1025−1044)

7 Contaldi C, Vafaee F, Nelson P C. Bayesian network hybrid learning using an elite-guided genetic algorithm. *Artificial Intelligence Review*, 2019, **52**: 245−272

8 Chow C, Liu C. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 1968, **14**(3): 462−467

9 Koopman R, Wang S H. Mutual information based labelling and comparing clusters. *Scientometrics*, 2017, **111**(2): 1157−1167

10 Pearl J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA: Morgan Kaufmann Publishers, 1988.

11 Jiang J K, Wang J Y, Yu H, Xu H J. Poison identification based on Bayesian network: a novel improvement on K2 algorithm via Markov blanket. In: Proceedings of the 2013 Advances in Swarm Intelligence. Lecture Notes in Computer Science. Berlin, Germany: Heidelberg: Springer, 2013. 173−182

12 Cooper G F, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 1992, **9**(4): 309−347

13 Chen X W, Anantha G, Lin X T. Improving Bayesian network structure learning with mutual information-based node ordering in the K2 algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 2008, **20**(5): 628−640

14 Ko S, Kim D W. An efficient node ordering method using the conditional frequency for the K2 algorithm. *Pattern Recognition Letters*, 2014, **40**: 80−87

15 Leray P, Francois O. BNT Structure Learning Package: Documentation and Experiments, Technical Report, Laboratoire PSI, 2006.

16 Faulkner E. K2GA: heuristically guided evolution of Bayesian network structures from data. In: Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining. Honolulu, HI, USA: IEEE, 2007. 18−25

17 Wu Y H, McCall J, Corne D. Two novel ant colony optimization approaches for Bayesian network structure learning. In: Proceedings of the 2010 IEEE Congress on Evolutionary Computation. Barcelona, Spain: IEEE, 2010. 1−7

18 Aouay S, Jamoussi S, Ben Ayed Y. Particle swarm optimization based method for Bayesian network structure learning. In: Proceedings of the 5th International Conference on Modeling, Simulation, and Applied Optimization. Hammamet, Tunisia: IEEE, 2013. 1−6

19 Liu Hao-Ran, Sun Mei-Ting, Li Lei, Liu Yong-Ji, Liu Bin. Study on Bayesian network structure learning algorithm based on ant colony node order optimization. *Chinese Journal of Scientific Instrument*, 2017, **38**(1): 143−150
(刘浩然, 孙美婷, 李雷, 刘永记, 刘彬. 基于蚁群节点寻优的贝叶斯网络结构算法研究. 仪器仪表报, 2017, **38**(1): 143−150)

20 Malone B, Kangas K, Järvisalo M, Koivisto M, Myllymäki P. Empirical hardness of finding optimal Bayesian network structures: algorithm selection and runtime prediction. *Machine Learning*, 2018, **107**(1): 247−283

21 Wang Shuang-Cheng, Leng Cui-Ping, Li Xiao-Lin. Learning Bayesian network structure from small data set. *Acta Automatica Sinica*, 2009, **35**(8): 1063−1070
(王双成, 冷翠平, 李小琳. 小数据集的贝叶斯网络结构学习. 自动化学报, 2009, **35**(8): 1063−1070)

22 Schwarz G. Estimating the dimension of a model. *The Annals of Statistics*, 1978, **6**(2): 461−464

23 Murphy K. The Bayes net toolbox for MATLAB. *Computing Science and Statistics*, 2001, **33**: 1024−1034

24 Wang J Y, Liu S Y. Novel binary encoding water cycle algorithm for solving Bayesian network structures learning problem. *Knowledge-Based Systems*, 2018, **150**: 95−110

25 Liu Hao-Ran, Lv Xiao-He, Li Xuan, Li Shi-Zhao, Shi Yong-Hong. A study on the fault diagnosis model of rotary kiln based on an improved algorithm of Bayesian. *Chinese Journal of Scientific Instrument*, 2015, **36**(7): 1554−1561
(刘浩然, 吕晓贺, 李轩, 李世昭, 史永红. 基于 Bayesian 改进算法的回转窑故障诊断模型研究. 仪器仪表学报, 2015, **36**(7): 1554−1561)

26 Liu H, Zhou S G, Lam W, Guan J H. A new hybrid method for learning Bayesian networks: separation and reunion. *Knowledge-Based Systems*, 2017, **121**: 185−197

王海羽 燕山大学信息科学与工程学院硕士研究生. 主要研究方向为贝叶斯网络, 动态贝叶斯网络, 人工智能, 进化算法, 故障诊断与预测.
E-mail: anderwwhy@outlook.com
(**WANG Hai-Yu** Master student at the College of Information Science and Engineering, Yanshan University. His research interest covers Bayesian network, artificial intelligence, evolutionary computation, and fault diagnosis and prediction.)

刘浩然 燕山大学信息科学与工程学院教授. 主要研究方向为贝叶斯网络, 人工智能, 无线传感器网络, 故障诊断与预测. 本文通信作者.
E-mail: liuhaoranysu125@163.com
(**LIU Hao-Ran** Professor at the College of Information Science and Engineering, Yanshan University. His research interest covers Bayesian network, artificial intelligence, wireless sensor networks, and fault diagnosis and prediction. Corresponding author of this paper.)

张力悦 燕山大学信息科学与工程学院硕士研究生. 主要研究方向为智能算法, 贝叶斯网络, 故障诊断与预测.
E-mail: zly15128506765@163.com
(**ZHANG Li-Yue** Master student at the College of Information Science and Engineering, Yanshan University. His research interest covers intelligent algorithms, Bayesian networks, and fault diagnosis and prediction.)

张春兰 燕山大学信息科学与工程学院硕士研究生. 主要研究方向为智能优化, 贝叶斯网络, 故障诊断与预测.
E-mail: 15076053886@163.com
(**ZHANG Chun-Lan** Master student at the College of Information Science and Engineering, Yanshan University. Her research interest covers intelligent optimization, Bayesian network, and fault diagnosis and prediction.)

刘 彬 燕山大学信息科学与工程学院教授. 主要研究方向为深度学习, 贝叶斯网络, 故障诊断与预测.
E-mail: liubin@ysu.edu.cn
(**LIU Bin** Professor at the College of Information Science and Engineering, Yanshan University. His research interest covers deep learning, Bayesian network, and fault diagnosis and prediction.)