

# 一种学习稀疏 BN 最优结构的改进 $K$ 均值分块学习算法

高晓光<sup>1</sup> 王晨凤<sup>1</sup> 邸若海<sup>1</sup>

**摘要** 目前贝叶斯网络 (Bayesian networks, BN) 的传统结构学习算法在处理高维数据时呈现出计算负担过大、在合理时间内难以得到期望精度结果的问题. 为了在高维数据下学习稀疏 BN 的最优结构, 本文提出了一种学习稀疏 BN 最优结构的改进  $K$  均值分块学习算法. 该算法采用分而治之的策略, 首先采用互信息作为节点间距离度量, 利用融合互信息的改进  $K$  均值算法对网络分块; 其次, 使用 MMPC (Max-min parent and children) 算法得到整个网络的架构, 根据架构找到块间所有边的可能连接方向, 从而找到所有可能的图结构; 之后, 对所有图结构依次进行结构学习; 最终利用评分找到最优 BN. 实验证明, 相比现有分块结构学习算法, 本文提出的算法不仅习得了网络的精确结构, 且学习速度有一定提高; 相比非分块经典结构学习算法, 本文提出的算法在保证精度基础上, 学习速度大幅提高, 解决了非分块经典结构学习算法无法在合理时间内处理高维数据的难题.

**关键词** 贝叶斯网络, 结构学习, 改进  $K$  均值算法, 分块学习

**引用格式** 高晓光, 王晨凤, 邸若海. 一种学习稀疏 BN 最优结构的改进  $K$  均值分块学习算法. 自动化学报, 2020, 46(5): 923–933

**DOI** 10.16383/j.aas.c180837

## A Block Learning Algorithm With Improved $K$ -means Algorithm for Learning Sparse BN Optimal Structure

GAO Xiao-Guang<sup>1</sup> WANG Chen-Feng<sup>1</sup> DI Ruo-Hai<sup>1</sup>

**Abstract** At present, the traditional structure learning algorithm of Bayesian networks (BN) shows the problem of excessive computational burden and difficulty in obtaining the desired accuracy in a reasonable time when processing high-dimensional data. In order to learn the optimal structure of sparse BN under high-dimensional data, this paper proposes a block learning algorithm with improved  $K$ -means algorithm for learning sparse BN optimal structure. The algorithm adopts the strategy of divide and conquer. Firstly, we use mutual information as the distance between nodes, and the improved  $K$ -means algorithm with mutual information is used to block the network. Secondly, the MMPC algorithm is used to obtain the skeleton of the whole network. According to the skeleton, the possible connection directions of all edges between the blocks are found, so that all possible graph structures are found; after that, structural learning is performed sequentially for all possible graph structures; finally, the best BN is found by using scoring function. Experiments show that compared with the existing block structure learning algorithm, the proposed algorithm not only learns the optimal structure of the network, but also improves the learning speed definitely. Compared with the non-blocking classical structure learning algorithm, the learning speed of the algorithm proposed in this paper is greatly improved on the basis of ensuring accuracy, which solves the problem that the traditional algorithms cannot process high-dimensional data in a reasonable time.

**Key words** Bayesian network (BN), structure learning, improved  $K$ -means algorithm, block learning

**Citation** Gao Xiao-Guang, Wang Chen-Feng, Di Ruo-Hai. A block learning algorithm with improved  $K$ -means algorithm for learning sparse BN optimal structure. *Acta Automatica Sinica*, 2020, 46(5): 923–933

贝叶斯网络 (Bayesian networks, BN) 是系统描述随机变量之间因果关系的网络模型<sup>[1]</sup>, 目前广

泛应用于基因分析<sup>[2–3]</sup>, 医疗诊断<sup>[4]</sup>、模式识别<sup>[5–6]</sup>、图像显著性检测<sup>[7]</sup> 等前沿研究领域, 已成为处理不确定性的主流模型. BN 学习分为参数学习和结构学习. 在实际分析数据时, 我们根据获得的数据可以知道各个节点间的因果关系, 并根据这些关系构建 BN 的结构, 即结构学习, 进而学得依赖强弱的条件概率, 即参数学习. BN 结构学习的目标是寻找到一个最优结构, 参数学习则是基于模型的正确结构, 如果没有正确的结构, 参数学习就失去了意义.

不同规模的 BN 结构学习的常用算法不尽相

收稿日期 2018-12-18 录用日期 2019-04-15  
Manuscript received December 18, 2018; accepted April 15, 2019  
国家自然科学基金 (61573285) 资助  
Supported by National Natural Science Foundation of China (61573285)  
本文责任编辑 胡清华  
Recommended by Associate Editor HU Qing-Hua  
1. 西北工业大学电子信息学院 西安 710129  
1. The School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710129

同<sup>[8]</sup>. 对小规模 BN 的结构学习, 目前学习算法主要分为近似学习和精确学习. 近似学习往往是通过施加约束找到相容模型或通过贪婪搜索找到评分最高的模型, 对规模较小的 BN 是可行的, 复杂度控制在合理的范围; 精确学习则是将 BN 结构学习转化成求评分最大值的数学问题, 最终求得网络的最优模型. 典型的精确学习算法有动态规划<sup>[9-10]</sup>、整数规划<sup>[11]</sup> 等. 精确学习算法与其他方法相比, 精度更高, 同时空间复杂度更高, 花费的计算资源更多, 但对于小规模 BN, 仍旧可以在合理时间内得到最优 BN. 然而从现实世界得到的数据往往是高维复杂的, 如何利用 BN 对高维数据建模具有现实意义.

对于大规模稀疏 BN 的结构学习, 目前有传统的非分块结构学习算法和较为新颖的分块结构学习算法两种. 传统的非分块结构学习算法直接从数据学习网络结构, 如传统的精确学习算法存在维数爆炸的问题, 在节点数大于 26 时, 已不能在合理时间内得到结果<sup>[12]</sup>; 传统的近似学习算法中有些算法仍旧适用, 但时间复杂度也较高.

此时可以考虑在网络学习之前先进行分块处理, 目前分块结构学习算法已有一些研究成果. Koller 和 Friedman 在文献 [13] 论证了为提高网络模型的泛化能力采用稀疏模型的优势和必要性. Li 等于 2014 年提出一种图分割的方法<sup>[14]</sup>, 能够有效减少大规模 BN 学习的时间消耗. 该方法使用爬山算法或贪婪搜索算法来进行分割后图的结构学习, 存在基于评分函数的学习算法固有的弱点: 易陷入局部最优. Liu 等于 2017 年提出的 SAR (Separation and reunion) 算法<sup>[15]</sup> 指出可以将学习大型有向无环图 (Directed acyclic graph, DAG, 是指任意一条边有方向, 且不存在环路的图) 的任务利用  $q$ -局部 ( $q$ -partial) 图和最小节点分离的策略分解成一些相对较小规模的 DAG, 并重新统一这些小 DAG 来构建整个 DAG. 该算法是当前最新的分块学习算法, 但其使用基于评分函数的学习算法对小 DAG 进行学习, 并未利用约束来减小搜索空间, 存在非精确学习固有的弱点.

以上算法旨解决节点数增多后, 实现对大规模 BN 的结构学习, 但学习算法依旧采用近似学习, 并未保证最终得到的是最优模型, 此时可能寻找到的只是局部最优网络, 或精度得到限制. 虽然大规模 BN 的分块近似学习取得了一些成果, 但在分块精确学习的研究方面, 尚未见相关报道.

网络的规模变大后, 精确学习面临的挑战就更多. 例如, 如何保证在网络规模变大, 且保持精度的情况下, 提升算法的速度; 如何在搜索复杂的候选空间时避免找到局部最优的网络, 使得到的网络一定

是全局最优的; 如何在网络规模变大后, 在不增大时间消耗的基础上, 提升精度等.

本文采用分离合并的策略, 提出一种新的分块学习算法, 力图降低大规模稀疏 BN 结构学习的复杂度, 在保证学得网络精度的基础上, 提升算法的速度, 同时也确保找到的是全局最优的网络.

首先, 必须找到一个适用于 BN 的分块算法, 本文通过对现有分块算法 (即聚类算法, 以下称为分块算法) 的分析, 发现密度聚类算法是根据样本的紧密程度确定分块的, 思路比较复杂, 不能很好地解释对 BN 分块的原理; 层次聚类算法<sup>[16-18]</sup> 采用贪心策略进行分块, 得到的不一定是全局最优结果, 而经典的原型聚类算法- $K$  均值算法<sup>[19]</sup> 在描述分块过程时直观清晰, 通过距离来判断是否聚为一块, 高效快速, 能很好地表达 BN 中各节点聚为一块的过程, 故本文考虑采用  $K$  均值算法进行分块.

但通过对  $K$  均值算法分析发现原算法采用的欧氏距离度量在衡量 BN 样本间距离时表示的是样本状态的相异度, 与我们研究的问题不一致, 而互信息是基于样本的概率值, 衡量的是样本间的相关性, 因此我们融合互信息对  $K$  均值算法进行了改进, 基于此本文提出了学习稀疏 BN 最优结构的改进  $K$  均值分块学习算法 (Optimal structure-improved K-means algorithm, OS-IKM algorithm). 该算法利用融合互信息的改进  $K$  均值算法对网络分块, 通过互信息对网络进行有效的分块, 得到网络的分块结构, 其次根据整个网络的架构找到块间所有边的可能连接方向, 从而找到所有可能的图结构, 再对所有图结构使用动态规划算法进行结构学习, 最终利用评分函数找到最优 BN. 本文不同于以上分块方法的是利用改进的  $K$  均值算法得到网络分块结构, 在结构学习部分采用动态规划算法, 实现对网络的精确学习.

本文的组织结构如下: 第 1 节简要介绍 BN 结构学习及网络聚类的相关理论; 第 2 节详细论述了学习稀疏 BN 最优结构的改进  $K$  均值分块学习算法, 包括提出算法的思路和算法中涉及的基本算法; 第 3 节用仿真实验验证本文算法的性能; 第 4 节简要总结了本文工作.

## 1 相关理论

### 1.1 BN 及其结构学习

BN 由有向无环图 (DAG) 和条件概率表 (Conditional probability tables, CPTs) 组成, 可定义为  $BN = (G, P)$ . 网络中的节点代表随机变量, 边代表节点之间的因果关系. 每个节点由

$P(\text{node}|\text{pa}(\text{node}))$  参数化,  $\text{pa}(\text{node})$  表示父节点.

在 BN 中, 相应的全概率公式为:

$$P(X_1, X_2, \dots, X_n) = P(X_1)P(X_2|X_1) \cdots P(X_n|X_1, X_2, \dots, X_{n-1}) \quad (1)$$

其中,  $X_1, X_2, \dots, X_n$  表示变量.

结构学习是在不知道网络结构  $G$  的情况下, 通过数据分析同时确定 BN 的结构  $G$  和参数  $P$ . 动态规划算法把 BN 学习过程分为寻找父节点图  $\text{parentGraph}$  和节点序图  $\text{orderGraph}$  两个阶段, 对节点  $X_i$ ,  $\text{parentGraph}(X_i)$  是由  $X_i$  和可能父节点  $\text{pa}(X_i)$  组成的有向图, 此处  $\text{pa}(X_i)$  是除过  $X_i$  的所有节点构成的幂集;  $\text{orderGraph}$  即是网络中所有节点按一定次序构成的幂集. 动态规划算法开始于零节点子网, 依次遍历评估  $\text{parentGraph}$  和  $\text{orderGraph}$ , 将节点递归添加到子网中, 直到所有节点被添加到网络中. 评估准则是最小描述长度 (Minimum description length, MDL) 评分函数:

$$MDL(V) = \min_{X \in V} \{MDL(V \setminus \{X\}) + \text{BestMDL}(X, V \setminus \{X\})\} \quad (2)$$

$$\text{BestMDL}(X, V \setminus \{X\}) = \min_{\text{pa}(X) \subseteq V \setminus \{X\}} MDL(X|\text{pa}(X)) \quad (3)$$

对于节点数  $n$  稍大的实际问题, 可能父节点  $\text{pa}(X)$  的个数将呈指数增长, 维数爆炸是动态规划算法的一大挑战.

## 1.2 聚类

对网络分块就是寻找网络的社团结构, 社团结构意味着网络中的节点可以分成组, 组中节点之间的连接更密集, 组之间的连接则稀疏. 而聚类是根据相似性, 将数据划分成有意义或有用的簇的无监督学习过程. 因此, 寻找社团结构的问题本质上是一个聚类问题.

聚类算法有三大类. 第一类是基于原型的聚类, 假设聚类结构可以用原型来表征, 通常用于真实世界的分类任务. 最著名的算法是  $K$  均值算法, 本文在此基础上进行改进得到改进的  $K$  均值算法, 完成了 BN 的分块任务; 第二类是基于密度的聚类, 该算法假定聚类结构可以由样本之间的紧密程度确定. 具有噪声的基于密度聚类算法 (Density-based spatial clustering of applications with noise, DBSCAN)<sup>[20-21]</sup> 是一种众所周知的密度聚类算法, 它基于域参数来描述样本分布的紧密程度; 第三类是层次聚类算法, 如采用自下而上策略的凝聚算法

或采用自顶向下策略的分裂算法. FastNewman 算法 (FN 算法)<sup>[22]</sup> 是一种基于贪婪思想的典型凝聚算法, 自下而上根据最大化模块度值的合并方法合并社团. 该算法主要应用于加权网络, 对于实际网络的研究具有重要意义, 因此我们将改进的  $K$  均值算法与该算法进行了对比实验.

## 2 学习稀疏 BN 最优结构的改进 $K$ 均值分块学习算法

### 2.1 改进的 $K$ 均值算法

在对大规模稀疏 BN 结构学习之前, 首先对网络中的节点进行分块. 通过对现有分块算法的分析, 发现  $K$  均值算法采用距离作为节点的相似性指标, 目标是使得各个块的平方和最小, 不断迭代, 直至块中心不再改变. 该算法的分块思路简单易懂, 具有可迁移性, 且相比其他分块算法高效快速, 故本文选用  $K$  均值算法对 BN 分块.

经过分析  $K$  均值算法, 发现  $K$  均值算法在对 BN 中节点分块时具有不合理性:  $K$  均值算法采用欧氏距离来标识两个标量元素的相异度, 而对 BN 中的两个节点, 衡量两个样本的状态之间的欧氏距离, 无法对应我们研究问题的物理意义;  $K$  均值算法在迭代时, 将各个聚类子集内所有数据样本的均值作为新的聚类中心, 并根据误差平方和准则函数来判断是否替换, 此时在 BN 中根据所有数据样本的均值找到的聚类中心不一定真实存在, 是一个虚拟的点, 这对于之后其他点向该点合并社团没有实际意义.

针对这两个不合理性, 本文对  $K$  均值算法进行了改进, 提出改进的  $K$  均值算法, 使之适应于 BN 的分块任务.

1) 采用互信息<sup>[23]</sup> 作为节点间距离度量. 互信息是一个联合分布中的两个信息的纠缠程度, 即  $I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \ln \frac{p(x, y)}{p(x)p(y)}$ . 互信息表述的是  $X$  和  $Y$  联合分布相对于假定  $X$  和  $Y$  独立情况下的联合分布之间的内在依赖性, 因此互信息是衡量 BN 中节点相关性的有效量度. 将互信息作为距离度量需满足下述基本性质:

非负性:  $d(X_i, X_j) \geq 0$ ; 由于互信息是从随机变量  $X$  和  $Y$  的整体角度出发, 并在平均意义上观察问题, 故平均互信息量不会出现负值,  $I(X; Y) \geq 0$ .

同一性:  $d(X_i, X_j) = 0$  当且仅当  $X_i = X_j$ ;  $I(X; Y) = 0$  当且仅当  $X$  和  $Y$  为独立随机变量. 当  $X$  和  $Y$  独立时,  $p(x, y) = p(x)p(y)$ , 因此  $\ln \frac{p(x, y)}{p(x)p(y)} = \ln 1 = 0$ .

对称性:  $d(X_i, X_j) = d(X_j, X_i)$ ;  $I(X; Y) = I(Y; X)$ , 由  $Y$  提取到的关于  $X$  的信息量与从  $X$

中提取到的关于  $Y$  的信息量是一样的,  $I(X; Y)$  和  $I(Y; X)$  只是观察者的立足点不同, 从另一个角度来看,  $I(X; Y) = H(X) + H(Y) - H(X, Y) = I(Y; X)$ .

直递性:  $d(X_i, X_j) \leq d(X_i, X_k) + d(X_k, X_j)$ .  
 $I(X; Y) \leq I(X; Z) + I(Z; Y)$ , 如果  $X \rightarrow Z \rightarrow Y$  构成马氏链, 当消息经过多级处理后, 随着处理器数目的增多, 输入消息与输出消息之间的平均互信息量趋于变小, 即  $I(X; Y) \leq I(X; Z)$ ,  $I(X; Y) \leq I(X; Z)$ .

由此可知, 互信息衡量的是 BN 中样本的相关性, 我们可以根据互信息表征的样本间的相关性对网络进行有效的分块, 使得 BN 的分块问题有实际意义, 两者的结合对 BN 分块效果有很大改进.

2) 借鉴  $K$  中心点聚类算法的思路, 随机选定聚类子集中非中心点作为新的聚类中心, 将中心点实体化. 当一个中心点被某个非中心点替代时, 除了未被替换的中心点外, 其余各点被重新分配. 然后反复用非中心点来代替中心点, 以优化聚类质量. 这一思路也可以减轻  $K$  均值算法对孤立点的敏感性.

聚类质量用一个代价函数来表示, 如果当前的中心点被非中心点所代替, 替换的总代价是所有非中心对象所产生的代价之和. 这里代价函数和互信息联系起来, 总代价就是聚类子集中各节点之间互信息的总和. 如果新聚类子集的互信息总和大于原聚类子集的互信息总和, 那么原聚类中心被替代.

根据以上介绍, 改进的  $K$  均值算法 (Improved  $K$ -means algorithm, IKM algorithm) 流程图如图 1, 算法的时间复杂度为  $O(nKl)$ , 其中  $n$  指的是数据集的大小,  $K$  为分块数,  $l$  为迭代次数, 一般  $K$  和  $l$  可认为是常量, 故改进的  $K$  均值算法的复杂度为  $O(n)$ .

## 2.2 Merge 函数

运行改进的  $K$  均值算法得到分块结果、及 MMPC 算法<sup>[24-25]</sup> 得到网络架构后, 此时的网络结构示意图如图 2.

图 2 中的无向网络即是 MMPC 算法得到的网络架构, 每一虚线圆表示一个分块, 由改进的  $K$  均值算法得到. 图中的无向网络亦是一个稀疏网络, 可以看到块内部节点间的边 (实线边) 多, 块内联系紧密; 块间联系的边 (虚线边) 少, 块间联系稀疏.

Merge 函数的作用是得到所有可能的图结构. 通过对块间的无向边假设方向, 依次存储可能的图结构以供之后的结构学习. 假设存在  $n$  条无向边 (虚线边), 边的箭头指向的可能组合有  $2^n$  种, 则可能的图结构有  $2^n$  种. 故 Merge 函数的时间复杂度为  $O(2^n)$ ,  $n$  表示块间联系的边数, 对于稀疏网络而言,

$n$  值较小, 故 Merge 函数的指数复杂度是允许的.

图 2 所示的网络可能的图结构有四种, 如图 3 所示.

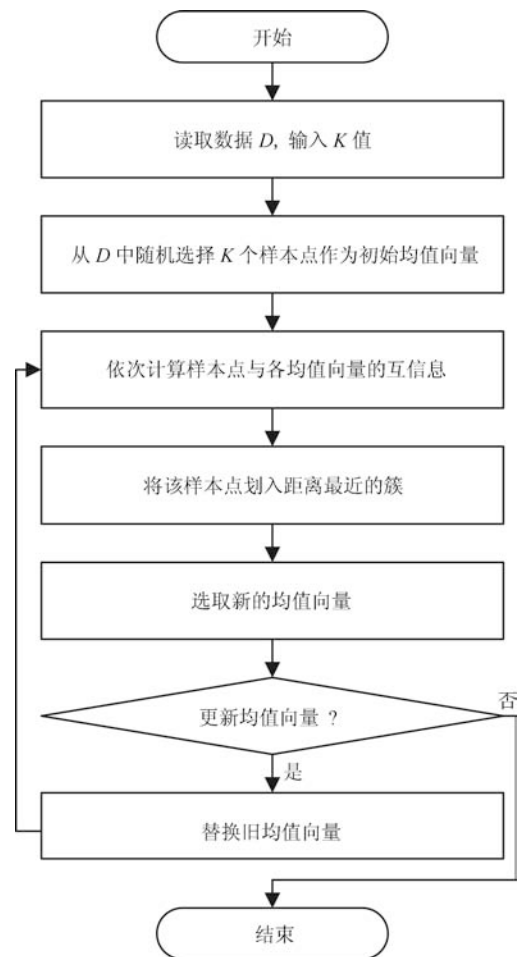


图 1 提出的改进  $K$  均值算法的流程图

Fig. 1 Diagram of the improved  $K$ -means algorithm

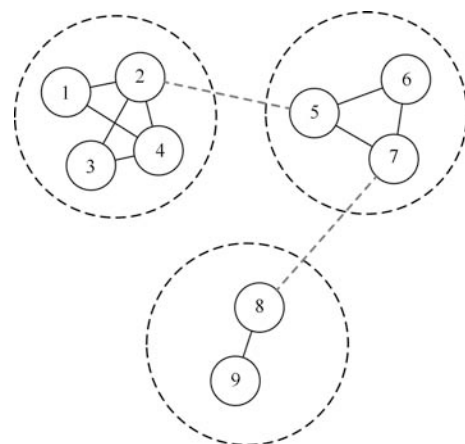


图 2 Merge 函数运行前网络结构示意图

Fig. 2 An example of a network structure before Merge function

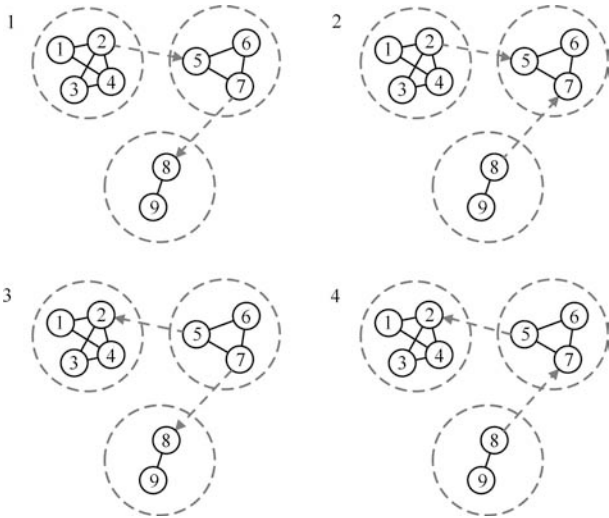


图 3 四种可能的图结构

Fig. 3 Four possible graph structures

### 2.3 剪枝后的动态规划算法

由以上算法, 找到了所有可能的图结构, 接下来使用动态规划算法依次学习所有可能的图结构, 利用评分找到最优的网络结构. 动态规划算法是一类应用较多的精确学习算法, 主要思想是把网络结构学习过程分为寻找父节点图和节点序图两个阶段, 对问题递归求解, 并使用 MDL 评分来衡量结构和数据集的匹配度. 每个节点对应一个父节点图, 图 4 示例节点  $X_1$  的父节点图, 包含所有其他节点的子集.

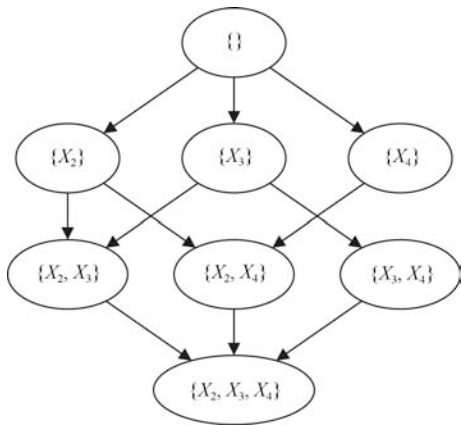


图 4 节点  $X_1$  的原始父节点图

Fig. 4 The original parentGraph of the node  $X_1$

本文利用 MMPC 算法得到的架构来修剪上图, 假设由 MMPC 算法得到节点  $X_1$  的邻居节点只有  $X_2$ 、 $X_3$ , 此时节点  $X_1$  的父节点图应剪枝掉包含  $X_4$  的所有集合, 如图 5 所示. 该方法可以降低运算复杂度.

对于  $n$  个节点的问题, 节点序图由  $2^n$  个子集作为节点组成. 图 6 表示的是 4 个节点的节点序图.

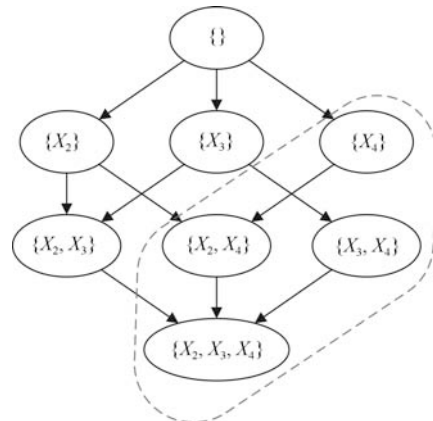


图 5 节点  $X_1$  的修剪后父节点图

Fig. 5 The modified parentGraph of the node  $X_1$

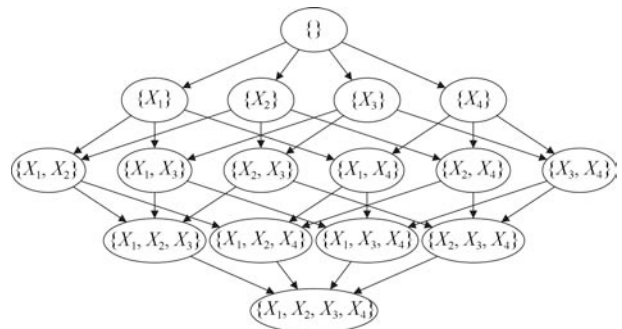


图 6 四个节点的节点序图

Fig. 6 The orderGraph of four nodes

父节点图和节点序图自然成层, 评估父节点图的某一层只需要前一层, 评估节点序图的某一层需要前一层节点序图和当前层的父节点图. 该层被评估后前一层就可以被删除, 而修剪过的父节点图减小了搜索空间, 再对分块得到的中小型网络进行结构学习, 算法可扩展性更大. 剪枝后的动态规划算法的时间复杂度为  $O(n2^n)$ ,  $n$  指的是数据集的大小.

### 2.4 学习稀疏 BN 最优结构的改进 $K$ 均值分块学习算法

总结前文的结构学习过程, 可将本文提出的学习稀疏 BN 最优结构的改进  $K$  均值分块学习算法 (OS-IKM algorithm) 归纳为如下四个步骤: 第一步, 由改进的  $K$  均值算法对网络分块; 第二步, 引用 MMPC 算法<sup>[24-25]</sup> 得到网络的无向图架构; 第三步, 调用 Merge 函数确定块与块之间的边, 并假设边的方向, 找到所有可能的图结构; 第四步, 对所有可能的图结构进行动态规划学习. 最终, 贝叶斯信息准则 (Bayesian information criterion, BIC) 评分找到最优网络. OS-IKM 算法的伪代码如下:

**算法 1.** OS-IKM algorithm

**Input:** data set  $D$ , number of clusters  $K$

**Output:** optimal BN  $N_G^*$  and score  $s^*$

1:  $C = \{C_1, C_2, \dots, C_k\} \leftarrow$  **IKM algorithm**  
( $D, K$ )

2: The skeleton of the network  $\leftarrow$  **MMPC Algorithm** ( $D$ )

3: All possible diagrams  $\delta^* \leftarrow$  **Merge** ( $C$ ,  
*The skeleton of the network*)

4: **for all**  $i = 1, 2, \dots, 2^m$

5:  $N_{G_i}^*$  and  $s_i^* \leftarrow$  **Dynamic Programming**  
( $D, \delta_i^*$ )

6: **end for**

7: optimal BN  $N_G^* \leftarrow \arg \max(s_i^*, N_{G_i}^*)$  and  
 $s^* \leftarrow \max(s_i^*, N_{G_i}^*)$

OS-**IKM** 算法得到的网络是全局最优的。

如图 7 所示意的四种可能的图结构, 对于第一种 ( $2 \rightarrow 5, 7 \rightarrow 8$ ) 的可能连接方式, 在对节点 5、6、7 组成的块动态规划学习时, 计算节点 5 的评分时需要考虑节点 2 作为父节点的评分, 同样在对节点 8、9 组成的块动态规划学习时, 计算节点 8 的评分时需要考虑节点 7 作为父节点的评分. 这样保证了经过分块没有遗漏掉在本块之外可能的父节点, 计算网络的评分是在全局范围内考量的. 通过动态规划算法学习到全局最优子网络结构<sup>[9]</sup> 后, 最终找到的网络应该是全局最优的。

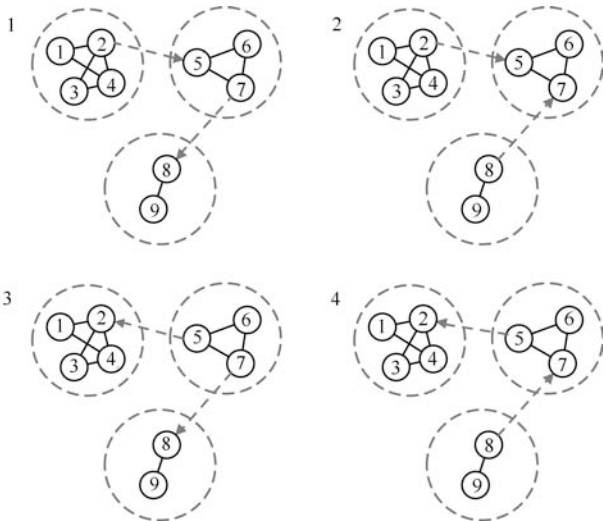


图 7 四种可能的图结构

Fig. 7 Four possible graph structures

### 3 OS-**IKM** 算法相关对比实验及分析

本文使用 BNrepository 库中四个网络及创建的网络 Mybnet1、Mybnet2 进行测试, 网络参数如表 1 所示。

表 1 实验中用到的 BN 的参数

Table 1 Details of Bayesian networks in the experiments

| 网络         | 节点数 | 边数 |
|------------|-----|----|
| Sachs      | 11  | 17 |
| Mybnet1    | 18  | 27 |
| Boerlage92 | 23  | 36 |
| Insurance  | 27  | 52 |
| Mybnet2    | 32  | 42 |
| Alarm      | 37  | 46 |

#### 3.1 改进的 $K$ 均值算法 (**IKM** 算法) 实验分析

为验证本文提出的改进  $K$  均值算法对 BN 的分块效果, 用模块度  $Q$  来量化聚类水平, 与第一节中提到的经典的社团挖掘算法 FN 算法在时间和精度上进行比较.  $Q$  是社团内部边的数量减去社团中所有节点连接的任何边的数量:

$$Q = \sum_i (e_{ii} - a_i^2) \quad (4)$$

其中,  $e_{ii}$  表示社团中边的比例,  $a_i$  表示与社团  $i$  内的点关联的所有边数量与边总数的比率.  $Q$  值越大, 即社团内部边越多, 社团与社团之间的边越少, 因此  $Q$  可以定量度量网络的聚类质量。

在样本量为 5000 的情况下, FN 算法和提出的 **IKM** 算法在不同网络下的  $Q$  值如图 8 所示。

根据图 8, 对于 Sachs 网络和 Boerlage92 网络, 由 **IKM** 算法得到  $Q$  值略低于 FN 算法, 这是节点数较少时, 网络连接相对致密, 分块的意义不明显, 由 **IKM** 算法中互信息计算得到的节点归属可能不唯一, 也就是说某一节点既可以归属于这一类, 也可以归属于另一个类, 具有很大不确定性, 导致精度不高. 如 Sachs 网络中节点 8 在真实结构中只与节点 1 有一条边连接, 属于孤立点, 此时两个算法的分块结果不同。

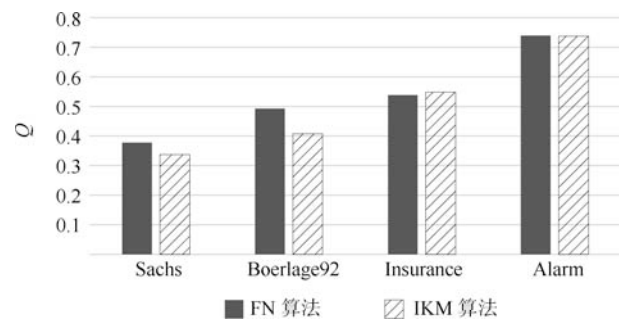


图 8 两个算法的  $Q$  值比较

Fig. 8 The  $Q$  of the two algorithms

对于较为复杂的 Alarm 网络, 两种算法的分块结果只有节点 24 的归属不一样, 节点 24 属于孤立点, 但此时它的归属对整个网络而言影响甚微. 由图

8, 两种算法得到的  $Q$  值相当, 可以看出网络规模变大后, IKM 算法得到了和 FN 算法一致的精度, 原因是在网络规模变大后, 孤立点对整个网络的影响变小, 此时更容易找到接近实际中心的聚类中心点, 算法精度明显提高. 这说明对大规模稀疏 BN, 分块思路是适用的.

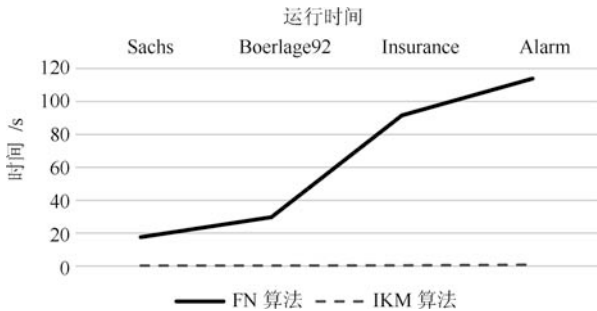


图 9 两个算法的运行时间比较

Fig. 9 The run time of the two algorithms

图 8 从量化准则上说明了本文提出的 IKM 算法可以达到和 FN 算法一致的精度, IKM 算法中用节点间互信息替代欧氏距离是合理的, 互信息可以作为节点相似性的度量, 这与前文分析是一致的. 通过模块度这一衡量标准, 可以看出  $Q$  值均保持在 0.3~0.8 之间, 根据经验来看, 此时网络分块效果较好, 这与我们观察到的结果是一样的.

本文在相同样本量情况下, 比较了两种算法的运行时间, 如图 9 所示. IKM 算法速度更快, FN 算法更耗时. 由于 FN 算法从网络架构开始, 从数据学习到网络架构花费了大量的时间, IKM 算法从数据直接计算互信息, 虽然计算互信息也花费了一些时间, 但相比 FN 算法减少了时间消耗.

在样本量为 5000 的情况下, 我们在 Boerlage92 网络上进行了多次分块实验, 并基于  $Q$  分析了本文提出的 IKM 算法的稳定性, 如图 10 所示. 实验中,  $Q$  一直保持在 0.4~0.5, 这表明虽然每个分块的结果略有偏差, 但分块的精度保持稳定.

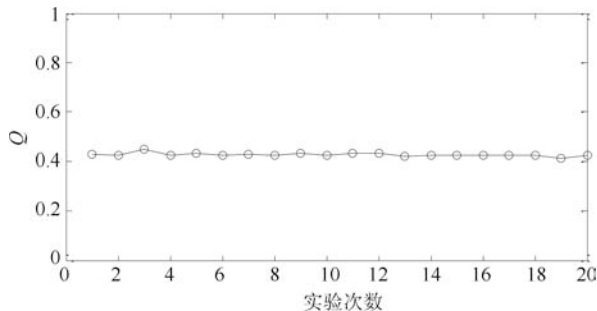


图 10 IKM 算法的稳定性

Fig. 10 The stability of the IKM algorithm

由以上分析, 本文提出的 IKM 算法能更快得到

与 FN 算法一致精度的结果, 并且是稳定的, 可以作为对网络分块的算法. 下面分析使用 IKM 算法分块的 OS-IKM 算法的性能.

### 3.2 OS-IKM 算法实验分析

下面验证 OS-IKM 算法是否达到了预期的时间优势. 在相同样本量 5000 的情况下, OS-IKM 算法及 Optimal structure-FastNewman (OS-FN) 算法、SAR 算法<sup>[13]</sup>、图分割 (Graph partitioning) 算法<sup>[12]</sup> 的运行时间如图 11 所示. OS-FN 算法是使用经典的社团挖掘算法 -FN 算法替代 IKM 算法对网络进行分块; SAR 算法是 Liu 等于 2017 年提出的一种新的分离合并算法; Graph Partitioning 算法是 Li 等于 2014 年提出一种图分割的方法, 包含 Max-min hill-climbing (MMHC) based on graph partitioning 和 Greedy search (GS) based on graph partitioning 两个算法, SAR 算法和 Graph partitioning 算法在上文都有提到. 该四类算法均是采用网络分块学习的思想, 故本文在此与其比较.

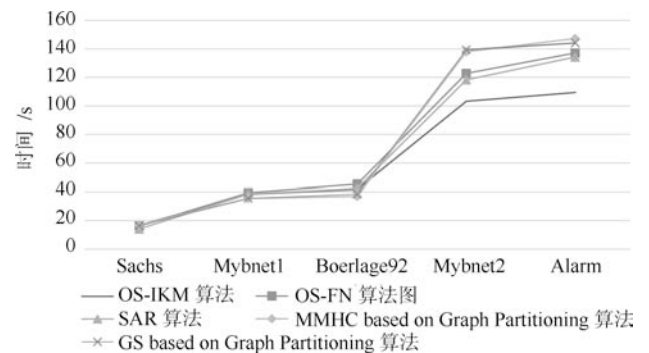


图 11 OS-IKM 算法与现有分块算法的运行时间对比

Fig. 11 The comparison of the run time between OS-IKM algorithm and other block algorithms

由图 11, 以上列出的五类算法时间复杂度均不高, 对于规模较大的 Alarm 网络, 运行时间大约都在 160 秒以下, 这说明了对于网络分块学习有利于减少运行时间. 由以上 FN 算法和本文提出的 IKM 算法的实验可以看到, IKM 算法的时间消耗更小, 两个算法最终得到的分块结果精度一致, 但略不相同, 这对之后的结构学习需要的时间有一定影响, 由图 11 也可以看到 OS-IKM 算法比 OS-FN 算法运行速度更快些. 这也说明分块算法的时间性能很大程度上影响了之后分块学习算法的效率, 验证了本文改进  $K$  均值算法的必要性.

对于 Sachs 网络、Mybnet1 和 Boerlage92 网络, 五类算法的运行时间相差不大, 这是因为对于中等偏小规模的网络, 网络分块学习的思想能够在一定程度上降低时间消耗, 但由于规模的限制, 降低

的效果并不明显. 而对于 Mybnet2 和 Alarm 网络, OS-IKM 算法的速度最快, 这是由于 OS-IKM 算法采用动态规划算法学习结构, 虽然动态规划算法存在维度爆炸的缺点, 但在分块降维后, 对每一小块的学习时间很少, 同时动态规划算法对网络分层递推学习, 效率很高, 本文又利用 MMPC 算法剪枝, 故整个算法的时间消耗不高; 而 SAR 算法采用基于评分的学习算法, 未约束搜索空间, 使得在网络增大后, 需要衡量的候选模型变多, 速度变慢; MMHC based on graph partitioning 是基于 MMHC 算法学习, GS based on graph partitioning 算法是基于 GS 算法学习, 这两类学习算法存在上文提到的变量众多时效率低的问题, 故对 Alarm 网络学习时呈现出时间上的劣势. 但对于每一小块而言, 结构学习的算法均可以满足变量不多的情况, 故这里时间上的差异不大. OS-IKM 算法相比于现有分块算法有一定时间优势.

下面本文将 OS-IKM 算法与非分块经典结构学习算法的运行时间也进行了比较, 如图 12. 动态规划算法<sup>[5]</sup> 是经典的精确学习算法, 爬山 (Hill-climbing, HC) 算法<sup>[23]</sup> 是一类较为经典的基于评分函数的学习算法. 本文与这两类算法比较是为了说明分块思想的时间优势.

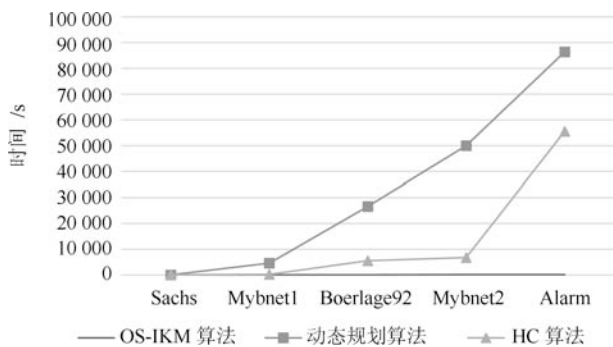


图 12 OS-IKM 算法与其他非分块经典结构学习算法的运行时间对比

Fig. 12 The comparison of the run time between OS-IKM algorithm and other traditional algorithms

由图 12, 随着网络规模的增大, OS-IKM 算法的运行时间一直很稳定, 而动态规划算法和 HC 算法的运行时间呈指数增长. 对 Mybnet2 和 Alarm 网络, 动态规划算法并没有在我们规定的时间 (1 天) 内运行出结果, 这里只是显示趋势, 这是由于动态规划算法对问题分层递推, 空间占用高, 对大于 23 个节点的网络已经很难得到结果. 而对 Alarm 网络, HC 算法用了大约 15 小时, HC 算法最费时的是计算候选模型的最大似然估计及评分, 模型的更新可以从前一步骤简单计算得到, 所以时间相对更快一些. 本文提出的 OS-IKM 算法对网络分块学习, 将

指数增长的时间控制在合理的范围, 网络分块的思想能大大降低时间复杂度, OS-IKM 算法相比非分块经典结构学习算法有很大的时间优势.

下面我们分析 OS-IKM 算法是否能达到和其他算法相同的精度, 结果如表 2 所示. 本文首先使用汉明距离进行比较, 下表中  $A(N_G^*)$  表示最终习得的网络与真实网络相比多余的边,  $M(N_G^*)$  表示最终习得的网络与真实网络相比缺失的边,  $I(N_G^*)$  表示最终习得的网络与真实网络相比反向的边, 汉明距离 = 多余边 + 缺失边 + 反向边, 即  $H(N_G^*)$ . 结构学习得出的网络与真实网络之间的汉明距离越小, 说明与真实网络相似度越高, 精确度越高.

对 Sachs 网络, SAR 算法习得的网络汉明距离最小, 本文提出的 OS-IKM 算法的精度处于中流, 对小网络的学习, 属于非分块经典结构学习算法能够解决的范畴, 动态规划算法和 HC 算法的学习精度较好; 对于其他网络, OS-IKM 算法的精度最高, 此时体现出精确学习相较于基于评分函数的学习算法和基于约束的学习算法在精度上的优势. 动态规划算法对 Mybnet2 和 Alarm 网络学习在规定时间内 (1 天) 没有得到结果, 故在表中用 “-” 表示.

不论对哪个网络, OS-IKM、OS-FN、MMHC based on graph partitioning 和 GS based on graph partitioning 四种算法得到的  $A(N_G^*)$  均为 0, 这是因为这些算法采用 MMPC 算法作为网络架构, 并在后续的学习中作为约束加入到网络学习中, 从而确保在学习某个节点的父节点时不会考虑其他多余的节点, 不会比真实网络多学到多余的边, 而 SAR 算法及其他非分块经典结构学习算法就会学得多余的边.

根据汉明距离, 本文提出的 OS-IKM 算法能够达到与其他算法一致的精度, 并在网络规模增大后有更大的潜力. 同时, 本文也比较了 OS-IKM 算法同其他算法的 BIC 评分, 如表 3. BIC 评分可以衡量习得的网络和真实网络的相似度. BIC 评分是在大样本前提下, 对边缘似然函数的一种近似. 模型结构的 BIC 评分, 记为  $BIC(N_G^*|D)$ , 即

$$\ln P(D|N_G^*) \approx \ln P(D|N_G^*, \theta^*) - \frac{d}{2} \ln m \quad (5)$$

第一项是模型  $N_G^*$  的优参数似然度, 度量结构  $N_G^*$  与数据  $D$  的拟合程度, 第二项是一个关于模型复杂度的罚项,  $d$  是参数向量  $\theta$  的独立参数的个数,  $m$  是样本量, 该罚项使得 BIC 评分有效地避免了过度拟合. BIC 评分选择既与数据拟合又比较简单的模型.

由表 3, 动态规划算法在 Mybnet2 和 Alarm 网络处数据缺失, 是因为对这两种网络学习在规定的时



表 2 汉明距离对比  
Table 2 The Hamming distances of the seven algorithms

|                   | OS-IKM 算法 | OS-FN 算法 | SAR 算法 | MMHC based on graph<br>partitioning 算法 | GS based on graph<br>partitioning 算法 | 动态规划算法 | HC 算法 |
|-------------------|-----------|----------|--------|--|--------------------------------------|--------|-------|
| (a) Sachs 网络      |           |          |        |  |                                      |        |       |
| $A(N_G^*)$        | 0         | 0        | 0      | 0                                      | 0                                    | 1      | 1     |
| $M(N_G^*)$        | 3         | 3        | 0      | 3                                      | 3                                    | 0      | 0     |
| $I(N_G^*)$        | 4         | 3        | 2      | 6                                      | 6                                    | 3      | 4     |
| $H(N_G^*)$        | 7         | 6        | 2      | 9                                      | 9                                    | 4      | 5     |
| (b) Mybnet1 网络    |           |          |        |  |                                      |        |       |
| $A(N_G^*)$        | 0         | 0        | 0      | 0                                      | 0                                    | 3      | 4     |
| $M(N_G^*)$        | 2         | 3        | 1      | 2                                      | 2                                    | 2      | 2     |
| $I(N_G^*)$        | 6         | 8        | 3      | 6                                      | 11                                   | 12     | 6     |
| $H(N_G^*)$        | 8         | 11       | 4      | 8                                      | 13                                   | 17     | 12    |
| (c) Boerlage92 网络 |           |          |        |  |                                      |        |       |
| $A(N_G^*)$        | 0         | 0        | 3      | 0                                      | 0                                    | 7      | 0     |
| $M(N_G^*)$        | 9         | 9        | 6      | 10                                     | 9                                    | 3      | 10    |
| $I(N_G^*)$        | 5         | 7        | 7      | 8                                      | 8                                    | 8      | 9     |
| $H(N_G^*)$        | 14        | 16       | 16     | 18                                     | 17                                   | 18     | 19    |
| (d) Mybnet2 网络    |           |          |        |  |                                      |        |       |
| $A(N_G^*)$        | 0         | 0        | 1      | 0                                      | 0                                    | –      | 0     |
| $M(N_G^*)$        | 5         | 6        | 5      | 7                                      | 7                                    | –      | 8     |
| $I(N_G^*)$        | 9         | 9        | 9      | 11                                     | 9                                    | –      | 6     |
| $H(N_G^*)$        | 14        | 15       | 15     | 18                                     | 16                                   | –      | 14    |
| (e) Alarm 网络      |           |          |        |  |                                      |        |       |
| $A(N_G^*)$        | 0         | 0        | 7      | 0                                      | 0                                    | –      | 8     |
| $M(N_G^*)$        | 9         | 11       | 4      | 8                                      | 11                                   | –      | 5     |
| $I(N_G^*)$        | 6         | 8        | 4      | 14                                     | 9                                    | –      | 12    |
| $H(N_G^*)$        | 15        | 19       | 15     | 22                                     | 20                                   | –      | 25    |

表 3 BIC 评分对比 ( $\times 10^4$ )  
Table 3 The BIC scores of the seven algorithms ( $\times 10^4$ )

|            | OS-IKM 算法 | OS-FN 算法 | MMHC based on Graph<br>Partitioning 算法 | GS based on Graph<br>Partitioning 算法 | 动态规划算法  | HC 算法   |
|------------|-----------|----------|--|--------------------------------------|---------|---------|
| Sachs      | -3.7004   | -3.7907  | -3.7038                                | -3.7813                              | -3.6616 | -3.6753 |
| Mybnet1    | -5.0766   | -5.0823  | -5.0833                                | -5.0853                              | -5.0946 | -5.0773 |
| Boerlage92 | -5.0474   | -5.0754  | -5.0853                                | -5.0773                              | -5.065  | -5.0758 |
| Mybnet2    | -9.1036   | -9.1300  | -9.1478                                | -9.1459                              | –       | -9.0947 |
| Alarm      | -6.521    | -6.7179  | -6.7889                                | -6.7486                              | –       | -6.8439 |

间内 (1 天) 没有得到结果. 对 Sachs 网络、Mybnet1 网络和 Boerlage92 网络, 六种算法的 BIC 评分值相当, 而根据表 2, 六种算法得到的汉明距离值有一定差别, 这是由于对比较简单的模型, BIC 评分更高, 而汉明距离衡量的是直观上和真实网络边的差别, 两者是从不同角度来衡量算法的精度. 对 Alarm

网络, 本文提出的 OS-IKM 算法的 BIC 评分值最高, HC 算法的 BIC 评分最低, 此时和表 2 中的结果是符合的, 这是因为 HC 算法极易陷入局部最优, 从而导致网络的精度不高, BIC 评分低, 而 OS-IKM 算法在全局范围内寻找最终网络, 可以保证全局最优性.

分析发现, OS-IKM 算法相较于非分块经典结构学习算法有明显的时间优势, 能大大降低时间复杂度; 相比现有分块结构学习算法, OS-IKM 算法有一定的时间优势, 如对 Alarm 网络大约减少了 25% 的时间消耗; 根据汉明距离和 BIC 评分, OS-IKM 算法达到了和目前现有算法一致的精度, 并有略微提升, 且保证最终习得的网络是全局最优网络. OS-IKM 算法在保证精度的前提下, 大幅提高学习速度, 对于经典动态规划算法不能在合理时间内得到学习结果的 Alarm 网络, 也能较快学习到全局最优结构, OS-IKM 算法是对大规模稀疏 BN 最优结构学习的一种可行算法.

#### 4 结束语

本文针对大规模稀疏 BN 最优结构的学习问题, 首先提出了 IKM 算法用于解决 BN 的分块问题, 实验显示 IKM 算法能更快地获得和 FN 算法一致精度的分块结果, 基于此, 本文提出了 OS-IKM 算法. 实验证明, 相比现有分块结构学习算法, OS-IKM 算法在保证习得精确结构的基础上, 有一定的时间优势, 如对 Alarm 网络, OS-IKM 算法在保证精度基础上减少 25% 的时间消耗; 相比非分块经典结构学习算法, OS-IKM 算法能在较短时间内达到与其一致的精度, 如对 Alarm 网络, 解决了经典动态规划算法在规定时间内没有得到结果的问题.

尽管本文算法在很多方面取得良好的结果, 但是仍然存在进一步提升的空间, 比如在对网络分块时, 仍然是根据经验预先给定分块数  $K$  值, 而分块数和网络的特性有很大关系, 在目前出现的分块算法中, 没有一个算法能够明确地告诉我们如何确定  $K$  值, 这将是我们的后续研究的方向.

#### References

- 1 Peal J. *Probabilistic Reasoning in Intelligent Systems*. Massachusetts: Morgan Kaufmann, 1988.
- 2 Soli R, Kaabi B, Barhoumi M, Maktouf C, Ahmed S. Bayesian phylogenetic analysis of the influenza-A virus genomes isolated in Tunisia and determination of potential recombination events. *Molecular Phylogenetics And Evolution*, 2019, **134**: 253–268
- 3 Stahl E A, Wegmann D, Trynka G, Gutierrez-Achury J, Do R, Voight B F, et al. Bayesian inference analyses of the polygenic architecture of rheumatoid arthritis. *Nature Genetics*, 2012, **44**(5): 483–489
- 4 Ehsani-Moghaddam B, Queenan J A, MacKenzie J, Birtwhistle R V. Mucopolysaccharidosis type II detection by naive Bayes classifier: An example of patient classification for a rare disease using electronic medical records from the Canadian Primary Care Sentinel Surveillance Network. *Plos One*, 2018, **13**(12): e0209018
- 5 Zheng Wen-Bo, Wang Kun-Feng, Wang Fei-Yue. Background subtraction algorithm with bayesian generative adversarial networks. *Acta Automatica Sinica*, 2018, **44**(5): 878–890  
(郑文博, 王坤峰, 王飞跃. 基于贝叶斯生成对抗网络的背景消减算法. *自动化学报*, 2018, **44**(5): 878–890)
- 6 Gu Xiao-Qing, Jiang Yi-Zhang, Wang Shi-Tong. Zero-order TSK-type fuzzy system for imbalanced data classification. *Acta Automatica Sinica*, 2017, **43**(10): 1773–1788  
(顾晓清, 蒋亦樟, 王士同. 用于不平衡数据分类的 0 阶 TSK 型模糊系统. *自动化学报*, 2017, **43**(10): 1773–1788)
- 7 Wang Song-Tao, Zhou Zhen, Qu Han-Bing, Li Bin. Bayesian saliency detection for RGB-D images. *Acta Automatica Sinica*, 2017, **43**(10): 1810–1828  
(王松涛, 周真, 曲寒冰, 李彬. RGB-D 图像的贝叶斯显著性检测. *自动化学报*, 2017, **43**(10): 1810–1828)
- 8 Han Shao-Jin, Li Jian-Xun. Structure learning algorithm for bayesian network based on probability density kernel estimation. *Computer Engineering & Applications*, 2014, **50**(15): 107–112  
(韩绍金, 李建勋. 基于密度核估计的贝叶斯网络结构学习算法. *计算机工程与应用*, 2014, **50**(15): 107–112)
- 9 Malone B, Yuan C, Hansen E A. Memory-efficient dynamic programming for learning optimal bayesian networks. In: *Proceedings of the Twenty-Fifth Association for the Advance of Artificial Intelligence Conference*. AAAI Press, 2011. 1057–1062
- 10 Yuan C, Malone B. Learning optimal bayesian networks: a shortest path perspective. *Journal of Artificial Intelligence Research*, 2013, **48**: 23–65
- 11 Bartlett M, Cussens J. Integer linear programming for the bayesian network structure learning problem. *Artificial Intelligence*, 2017, **244**: 258–271
- 12 Singh A P, Moore A W. Finding optimal bayesian networks by dynamic programming. Technical Report CMU-CALD-05-106, School of Computer Science, Carnegie Mellon University, USA, 2005.
- 13 Koller D, Friedman N. *Probabilistic Graphical Models*. Cambridge, MA: MIT Press, 2009.
- 14 Li S H, Zhang J, Huang K H, Gao C X. A graph partitioning approach for bayesian network structure learning. In: *Proceedings of the 33rd Chinese Control Conference*. Nanjing, China: IEEE, 2014
- 15 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
- 16 Newman M E J, Girvan M. Finding and evaluating community structure in networks. *Physical Review E*, 2004, **69**(2): 026113
- 17 Newman M E J. Modularity and community structure in networks. In: *Proceedings of the 2006 American Physical Society March Meeting*. American Physical Society, 2006. 8577–8582
- 18 Zhang W, Zhang G X, Chen X H, Liu Y Q, Zhou X M, Zhou J L. DHC: a distributed hierarchical clustering algorithm for large datasets. *Journal of Circuits, Systems and Computers*, 2019, **28**(4): 1950065.1-1950065.26

- 19 Zhou Zhi-Hua. Machine Learning. Tsinghua University Press, 2016.
- 20 Birant D, Kut A. ST-DBSCAN: an algorithm for clustering spatial-temporal data. *Data and Knowledge Engineering*, 2007, **60**(1): 208–221
- 21 Viswanath P, Pinkesh R. I-DBSCAN: a fast hybrid density based clustering method. In: Proceedings of IEEE 18th International Conference on Pattern Recognition International Conference on Pattern Recognition. Hong Kong, China: IEEE Computer Society, 2006. 912–915
- 22 Newman M E J. Fast algorithm for detecting community structure in networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 2004, **69**(6): 066133.
- 23 Vinh N X, Chetty M, Coppel R, Wangikar P P. GlobalMIT: learning globally optimal dynamic bayesian network with the mutual information test criterion. *Bioinformatics*, 2011, **27**(19): 2765–2766
- 24 Konstantinos T, Vincenzo L, Sofia T, Tsamardinos I. On scoring maximal ancestral graphs with the max-min hill climbing algorithm. *International Journal of Approximate Reasoning*, 2018
- 25 Tsamardinos I. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 2006, **65**(1): 31–78



**高晓光** 西北工业大学电子信息学院教授. 1989 年获得西北工业大学飞行器导航与控制系统博士学位. 主要研究方向为贝叶斯网络和复杂系统建模与分析. 本文通信作者.

E-mail: cxg2012@nwpu.edu.cn

(**GAO Xiao-Guang** Professor at the School of Electronics and Informa-

tion, Northwestern Polytechnical University. She received

her Ph.D. degree in aircraft navigation and control system from Northwestern Polytechnical University in 1989. Her research interest covers Bayesian networks, modeling and analysis of complex systems. Corresponding author of this paper.)



**王晨凤** 西北工业大学电子信息学院硕士研究生. 2017 年获得西北工业大学学士学位. 主要研究方向为贝叶斯网络和数据挖掘.

E-mail: chen-cc@mail.nwpu.edu.cn

(**WANG Chen-Feng** Master student at the School of Electronics and Information, Northwestern Polytechnical

University. She received her bachelor degree from Northwestern Polytechnical University in 2017. Her research interest covers Bayes networks and data mining.)



**邸若海** 西北工业大学电子信息学院博士后. 2016 年获得西北工业大学系统工程专业博士学位. 主要研究方向为小数据集条件下贝叶斯网络结构学习和参数学习. E-mail: diruohai@nwpu.edu.cn

(**DI Ruo-Hai** Postdoctoral at the School of Electronics and Information, Northwestern Polytechnical University.

He received his Ph.D. degree in system engineering from Northwestern Polytechnical University in 2016. His research interest covers structure and parameter learning of Bayesian networks from small data set.)