基于集成学习和反馈策略的贝叶斯网络结构学习

王守会 覃 飙
(中国人民大学信息学院 北京 100872)

摘 要 为解决单一结构学习算法中普遍存在的学习效果差、易陷入局部最优等问题,本文通过引入最大信息系数 *MIC*(Maximal Information Coefficient)和集成学习思想,提出了一种新的贝叶斯网络结构学习算法.该算法首先利用 *MIC* 确定节点间的依赖关系并得到初始网络,接着使用 Bootstrap 方法对数据样本进行采样获得若干样本集.依次使用每个样本集,在给定初始网络的基础上,通过 BDe 评分和禁忌搜索算法训练生成贝叶斯网络并将其用邻接矩阵表示.进而利用集成学习策略,根据得到的若干邻接矩阵计算每条边的权重,并通过设定权重阈值得到集成学习结果.根据集成学习结果利用反馈策略更新初始网络并进入下一次迭代,经过不断迭代最终得到贝叶斯网络结构.最后在 7 种不同大小的标准数据集中进行实验,计算 *F*1 值(*F*1-score)、汉明距离 *HD*(Hamming Distance)和学习到的正确边数(*TP*),并与其它算法进行对比分析.结果表明本文提出的算法在可行性、有效性和普适性上更优.

关键词 集成学习;贝叶斯网络;结构学习;反馈策略;最大信息系数
中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2021.01051

Bayesian Network Structure Learning by Ensemble Learning and Feedback Strategy

WANG Shou-Hui QIN Biao
(*School of Information, Renmin University of China, Beijing* 100872)

Abstract The Bayesian network is an effective tool to express uncertain information in the field of artificial intelligence. It is widely used in various fields. Bayesian network learning mainly includes structure learning, parameter learning and inference. The Bayesian Network Structure Learning (BNSL) is the key and difficulty of Bayesian network research. The task of BNSL is an NP-Hard problem. At present, the methods for BNSL mainly include constraint-based structural learning, score-based structural learning and hybrid-based structural learning. By analyzing the state-of-the-art structural learning methods, we found that they not only have bad efficient and effective but also easily fall into local optimality. In order to improve the accuracy and reliability of structural learning, we propose a new algorithm called Ensemble-Feedback Bayesian Network Structure Learning (EFBNSL) in this paper. The ensemble learning strategy is used to reduce the number of invalid edges (e.g. multi-edge, less-edge, and reverse-edge) and the feedback strategy is used to update the initial network and enhance the quality of initial network. This algorithm is mainly divided into the following two steps. In the first step, we build the initial network. We employ Maximal Information Coefficient (*MIC*) to determine whether there is an edge between the random nodes, and construct the initial network (directed graph) according to edge-adding rules. In the second step, we use the Bootstrap method to sample n samples from the data sets. We use

收稿日期:2019-09-09;在线发布日期:2020-03-10.本课题得到国家自然科学基金(61772534)资助.**王守会**,博士研究生,主要研究方向为贝叶斯网络、知识库问答.E-mail:wsh_inf@ruc.edu.cn.**覃 飙**(通信作者),博士,副教授,中国计算机学会(CCF)会员,主要研究方向为贝叶斯网络、数据挖掘.E-mail:qinbiao@ruc.edu.cn.

the initial network bn_{k-1} which represents the network after kth iteration to initialize the Tabu search algorithm, and combine the BDe score function for Bayesian network structural learning from each sample. As a result, we obtain n Bayesian networks that are represented n matrices. Then, we calculate the weight of each edge by the function of ensemble strategy W according to n matrices. We determine the directed edge by setting the threshold θ, and generate the Bayesian network BN_k, where BN_k represents the Bayesian network after kth iteration. We use the BN_k to update the initial network bn_{k-1} according to the update condition C and use the updated bn_k for the next iteration. We obtain the optimal Bayesian network, which is the network that our EFBNSL algorithm outputs when it is satisfied with the termination condition by ensemble learning and the feedback strategy in an iterative way. Finally, we perform experiments using seven benchmarks with different sizes. We calculate the $F1$-score, Hamming Distance and the number of correct edges of the Bayesian network and compare them with other state-of-the-art algorithms. The result shows that the EFBNSL algorithm proposed in this paper is superior to other algorithms in terms of effectiveness and generalization ability. After analysis, we conclude that the success of this algorithm has the following abilities: First, we use MIC to construct the initial network structure directly, it easier than the previous method, which first obtains the undirected edges, then eliminates the loops, and finally decides their directions. Second, we propose a function of ensemble strategy, which is used to reduce the number of invalid edges that may exist in the Bayesian network. Finally, we employ the feedback strategy to adjust the edge directions of the initial network, and improve the quality of the initial network, and then we can obtain the optimal Bayesian network by iterative optimization.

Keywords ensemble learning; bayesian network; structural learning; feedback strategy; maximal information coefficient

1 引 言

贝叶斯网络(Bayesian Network,BN)是概率图模型的一种,能够对不确定知识进行有效推理和分析,是人工智能领域研究的一个重要分支.贝叶斯网络有着坚实的理论基础,并在交通[1]、工业生产[2]、医疗[3]、农业[4]等行业得到了广泛应用.贝叶斯网络学习主要包括结构学习、参数学习和知识推理三个方面,其中结构学习是贝叶斯网络学习的重点和难点.

结构学习的目标是学习得到一个与样本数据集拟合度最高的网络结构,事实证明这是一个 NP-Hard 问题[5].贝叶斯网络结构学习分为完备和非完备数据集下的结构学习.在完备数据集下进行结构学习的主要方法有:(1)基于依赖统计分析的方法;(2)基于评分搜索的方法;(3)综上两种方法的混合学习策略.基于依赖统计分析的方法,其基本思想是利用互信息或条件互信息对变量之间的依赖关系进行测试,以确定变量之间的条件独立性,并据此得到一个有向图.由于马尔可夫等价类具有相同的条件独立性(如 $X\rightarrow Z\rightarrow Y$, $X\leftarrow Z\leftarrow Y$, $X\leftarrow Z\rightarrow Y$),使得该方法不能有效识别数据集中可能存在的马尔可夫等价类,因而无法确定有向边.该情况在高维网络结构中更常见,因此该方法更适用于稀疏网络的结构学习,且对于结构复杂的网络其计算复杂度会大幅度增加,学习效果也会变差.为了提高结构学习效率和独立性测试效率,一般采用 D-分离树(D-Separators)对网络结构进行分解,形成若干子网络再进行结构学习[6].基于评分搜索的结构学习方法采用启发式搜索策略,其关键在于评分函数和搜索算法的选择.评分函数用于评价学习到的网络与实际网络的相似程度,常用的评分函数有 BIC[7]、MDL[8]、BDe[9]等.搜索算法决定了在特定网络空间中的搜索策略,当评分函数确定后,通过搜索算法寻找最佳的网络结构.经典的搜索算法为 K2 算法[10],K2 算法需要节点顺序和最大父节点数作为先验知识且易于陷入局部最优.上述两种结构学习方法各有优劣,因此综上两种方法的混合学习策略成为研究热点.经典的混合学习方法为最大最小爬山算

法[11](Max-Min Hill-Climbing,MMHC),其核心思想是:首先通过局部发现算法(Max-Min Parents and Children,MMPC)确定贝叶斯网络框架(无向图),然后通过贝叶斯评分算法和爬山算法相结合来确定边的方向,并得到最终的贝叶斯网络结构.为了解决初始网络为空时,贪婪等价搜索算法(Greedy Equivalent Search,GES)易陷入局部最优,Zhang 等人[12]提出了改进的 GES 算法.该算法首先通过条件独立性测试和最大信息系数确定变量间的独立性并构建初始网络,进而将初始网络作为种子利用 GES 算法进行结构学习.但在构建初始网络时,分为构建无向图、消除环结构和定向三个部分,且初始网络的质量直接决定了最终贝叶斯网络的好坏.

对现有的结构学习方法分析发现,结构学习的可行性、准确性有待提高.为此,Cai 等人[13]针对依赖统计分析方法中通常得到的是贝叶斯网络的马尔可夫等价类,由于无向边的存在而无法进行有效的因果推断的问题,提出了一种融合学习策略,有效地减少了无向边的个数.但作者使用的是基于约束的 PC、IAMB 和 MMPC 3 种方法的融合,因此该方法对于大型网络其学习效率会降低.为进一步提高贝叶斯网络结构学习的有效性和学习效率,本文提出了一种基于集成思想的结构学习算法(Ensemble-Feedback Bayesian Network Structure Learning,EFBNSL).该算法主要分为以下 2 个步骤,(1)初始网络构建.利用最大信息系数 *MIC*(Maximal Information Coefficient)对边进行检测,确定变量之间是否存在边,并根据加边策略确定初始网络(如图 1 所示);(2)集成学习.利用 Bootstrap 方法从原始数据选出 n 份训练样本.依次将每份训练样本在给定初始网络 bn_{k-1} 的基础上(bn_{k-1} 表示第 k 轮迭代后的初始网络),利用 BDe 评分函数和禁忌搜索算法(Tabu Search)训练生成贝叶斯网络,并将该网络进行矩阵化表示.利用最终得到的 n 个矩阵,根据集成策略函数 W 计算每条边的权重,通过设定阈值 θ 确定有向边,并得到贝叶斯网络 BN_k(BN_k 表示第 k 轮迭代后的贝叶斯网络).根据更新条件 C,利用生成的贝叶斯网络 BN_k 对初始网络 bn_{k-1} 进行更新,得到新的初始网络 bn_k,并将其用于下一轮迭代,循环迭代直到满足停止条件时退出循环,得到最优贝叶斯网络.

本文的主要贡献如下:

(1)提出集成策略函数计算边的权重,利用集成学习方法减少贝叶斯网络中可能存在的多边、少边和反边情况.

(2)根据节点间的 *MIC* 值直接确定有向边,并得到初始网络,相较于文献[12]中先构造无向图再消除环结构最后进行定向的操作更加简便.

(3)提出反馈更新策略,根据集成学习结果对初始网络结构(边的方向)进行动态调整,以保证先验知识的有效性.因为较好的初始网络能够提高算法的搜索效率.

(4)在多个大小不同的标准网络上进行实验,验证了本文所提方法用于不同大小、不同复杂程度的贝叶斯网络结构学习具有较好的泛化能力.

2　相关工作

2.1　基于依赖统计分析的方法

在基于依赖统计分析的方法中,SGS 算法[14]是最早的结构学习算法,该算法主要根据各节点间的条件独立性来确定最终的网络结构,其学习效率会随着节点数的增加呈指数增长.为此,Spirtes 等人[15]对 SGS 算法进行改进,提出了 PC 算法,该算法能够有效地从稀疏模型中构建贝叶斯网络.Cheng 等人[16]将信息论用于贝叶斯网络结构学习,通过互信息来确定节点间的相关关系,将整个结构学习过程分为 Drafting、Thickening 和 Thinning 三个阶段,提出了 TPDA 算法.

此类算法需要依次判断各节点间的条件独立性,其最大的弊端是随着网络复杂度的增加,时间复杂度呈指数增长.因此该类算法比较适用于稀疏网络.对于此,洪英汉[17]在 PC 算法的基础上,加入基于最大依赖性、最小冗余度的互信息加速模块,提出了 FCNSLA 算法,在特征选择和相邻节点搜索方面都做了优化.

2.2　基于评分搜索的方法

在基于评分搜索的结构学习方法中,目前的算法主要包括优化评分函数、改进搜索算法或优化搜索空间.Constantinou[18]提出一种新的用于评估结构学习算法性能的评分函数(BSF 评分),该方法相较于传统的评分函数能够提供更公平的评估.由于 K2 算法需要已知节点次序作为先验知识,因此 Song 等人[19]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.Chen 等人[20]针对 K2 算法的弊端,先通过互信息确定变量节点的顺序,再利用 K2 算法进行结构学习.Liu 等

人[21]针对 K2 算法过度依赖节点顺序,而遗传算法用于节点序寻优效率差的问题,提出了一种直接对节点序进行评分搜索的结构学习算法. Fan 等人[22]在给定评分函数的情况下,通过优化搜索空间来改进结构学习算法. Gheisari 等人[23]利用学习自动机(Learing Automata)来指导搜索方案,使算法能够在可接受的时间范围内找到最优的网络结构. Liu 等人[24]将改进的鲸鱼优化策略用于贝叶斯网络结构学习. Askari 等人[25]利用布谷鸟搜索算法建立一个类似于布谷鸟的有向无环图种群,每个图都有一个表示其适应度的分数,通过不断重复该算法,直至找到最佳或可接受的网络结构. 近年来基于仿生学理论的搜索优化方法被广泛用于贝叶斯网络结构学习,如人工蜂群算法[26]、蚁群算法[27]、遗传算法[28-29]、人工免疫算法[30]等.

2.3 混合学习方法

随着网络结构复杂度的增加,对于依赖统计分析的结构学习方法其算法复杂度呈指数增加,且该方法对数据要求较高;对于评分搜索的结构学习方法其搜索空间会变得非常巨大,而在巨大的搜索空间中寻找最佳的网络结构是 NP-hard 问题. 为了克服上述两类方法的缺陷,研究者们将这两类方法进行结合提出了混合学习方法. 该类方法先通过独立性测试来降低搜索空间的大小,再利用评分搜索方法来寻找最优的网络结构. 例如一些学者将最大信息系数与贪婪算法相结合[12,31],提出了一种改进的贝叶斯网络结构学习算法. Contaldi 等人[28]提出了一种新的混合结构学习算法,该算法利用节点间相互依赖关系来降低搜索空间的复杂度,并引入遗传算法对可能的结构进行有效搜索.

3 基本概念

3.1 贝叶斯网络

贝叶斯网络又称信念网(Belief Networks)由 Pearl 在 1986 年提出[32],是一种结合概率论和图论的不确定知识表示和推理模型. 贝叶斯网络通常表示为 $BN=(G,P)$,其中 $G=(V,E)$表示有向无环图结构(Directed Acyclic Graph,DAG),用来刻画节点之间的依赖关系;$V=\{x_1,x_2,\cdots,x_n\}$为节点集;E为有向边集,表示节点间的直接依赖关系;P为节点间的概率分布,表示节点间依赖关系的强弱,一般用条件概率表(Conditional Probability Table,CPT)来描述. 贝叶斯网络在属性表达上包含了一个条件独立性假设,即在给定父节点集的条件下,节点 x_i 独立于除父节点集之外的所有非后代节点.

定义 1. 节点集 $V=\{x_1,x_2,\cdots,x_n\}$的联合概率分布表示为

$$P(x_1,x_2,\cdots,x_n)=\prod P(x_i\mid pa(x_i)),$$

式中,对于任意节点 $x_i\in V$,其条件概率表示为 $P(x_i\mid pa(x_i))$,其中 $pa(x_i)$是 x_i的父节点.

贝叶斯网络结构学习的目标是通过学习得到与样本数据集拟合度最高的网络结构. 评分函数通常被用于判断贝叶斯网络的好坏. 本文采用基于贝叶斯统计思想的 BD(Bayesian Dirichlet)评分函数.

定义 2. BD 评分函数利用先验知识和数据寻求后验概率最大的网络结构. 它假设数据服从狄里克雷分布(Dirichlet Distribution),其定义如下:

$$F_{\mathrm{BD}}(G\mid D)=\log(P(G))+\sum_{i=1}^{n}\sum_{j=1}^{q_i}\left[\log\left(\frac{\Gamma(\eta_{ij})}{\Gamma(\eta_{ij}+N_{ij})}\right)+\sum_{k=1}^{r_i}\log\left(\frac{\Gamma(\eta_{ijk}+N_{ijk})}{\Gamma(\eta_{ijk})}\right)\right],$$

其中,$P(G)$表示有向无环图 G 的先验概率. n 表示节点个数,q_i表示第 i 个节点的父节点数,r_i表示第 i 个节点的取值个数. N_{ijk}表示当节点 $x_i=k$,$pa(x_i)=j$ 时的样本个数,因此有 $N_{ij}=\sum_{k=1}^{r_i}N_{ijk}$. $\eta_{ij}=\sum_{k=1}^{r_i}\eta_{ijk}$,$\eta_{ijk}$表示给定网络结构的 Dirichlet 先验分布的超参数. 一般通过 $\eta_{ijk}=\eta/r_iq_i$得到超参数 η_{ijk},此时即为 BDe 评分.

3.2 最大信息系数

最大信息系数(MIC)是衡量变量 X 和 Y 之间关联程度的一种方式[33]. MIC 能够更准确地描述变量之间可能存在的关联关系,且不受数据分布的影响,不限于特定的关联函数类型,更具公平性和广泛性[31].

定义 3. 将坐标平面按 i 行和 j 列划分为(i,j)网格 G(未必等宽),其中 $i*j<n^{0.6}$. 使得包含变量 X 和 Y 的有限数据集 D 中的所有变量都落入 G 中,计算网格 G 下的互信息 MI(Mutual Information)I_G,其定义如下:

$$I(X;Y)=\int_Y\int_X p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right)\mathrm{d}x\mathrm{d}y.$$

定义 4. 已知数据集 D 中的两个节点变量 X 和 Y,其 X 和 Y 特征矩阵 $\boldsymbol{M}(D)$的计算公式定义如下:

$$\boldsymbol{M}(X,Y|D)_{i,j}=\frac{I^*(X,Y,D,i,j)}{\log\min(i,j)},$$

其中,$I^*(X,Y,D,i,j)$表示数据集 D 在网格 G 下变量 X 和 Y 的最大互信息值,即 $I^*(X,Y,D,i,j)=\max I(X,Y,D|_G)$.

定义 5.　在坐标平面内两节点变量 X 和 Y 之间的最大信息系数定义如下:

$$\boldsymbol{M}(X,Y|D)=\max_{i*j<n^{0.6}}\boldsymbol{M}(X,Y|D)_{i,j}.$$

由于随机变量之间的互信息具有对称性,因此 MIC 也具有对称性,即 $MIC(X,Y)=MIC(Y,X)$.

3.3　集成学习

集成学习(Ensemble Learning)是一种新的机器学习方法,被广泛地用于分类和回归问题,是机器学习中的研究热点之一.集成学习是指利用多个相同或不同的学习算法,通过某种组合来解决同一问题.研究表明,通过集成学习将多个模型进行结合得到的模型往往好于单一模型的学习效果,且可获得比单一模型更显著的泛化性能[34-35].

集成算法中最经典的为 Boosting[36] 和 Bagging[37] 集成学习. Boosting 算法的核心思想是对原始数据集进行有放回抽样来产生训练集并进行学习器的训练. Bagging 算法的核心思想是利用可重复取样技术(Bootstrap)采样产生多个数据集,并分别在这些数据集上训练学习器,进而对多个学习器进行组合来得到更具稳定性的学习器.贝叶斯网络结构学习的主要目的是确定节点间的有向边,通过 Bagging 学习可以有效减少贝叶斯网络在生成过程中可能存在的多边、反边问题,使得学习结果更具稳定性和可靠性.

4　EFBNSL 算法构建

4.1　初始网络构建

利用 MIC 对节点间的依赖关系进行检测.根据定义可知变量间的 MIC 值与其依赖程度正相关,即如果变量 X 和 Y 之间的 MIC 值越大,说明变量 X 和 Y 之间的依赖程度越高,在网络结构中表示它们之间直接或间接相连;若变量 X 和 Y 之间的依赖程度越低,则 MIC 值越小;当变量 X 和 Y 的 MIC 值为 0 时,表示变量之间相互独立,在网络结构中表示它们之间没有边相连.下面以 Asia 数据集为例说明初始网络的构建过程.计算数据集中任意变量之间的 MIC 值如表 1 所示,并据此进行初始网络的构建.

表 1　Asia 数据集各变量之间的 *MIC* 值

	A	T	S	L	B	E	X	D
A		0.003	0.002	0.001	0.000	0.000	0.000	0.000
T	**0.003**		0.000	0.001	0.001	0.024	0.019	0.000
S	0.002	0.000		0.043	0.046	0.037	0.016	0.022
L	0.001	0.001	**0.043**		0.001	**0.306**	0.213	0.020
B	0.000	0.001	**0.046**	0.001		0.001	0.000	**0.369**
E	0.000	**0.024**	0.037	**0.306**	0.001		**0.238**	0.019
X	0.000	0.019	0.016	0.213	0.000	0.238		0.007
D	0.000	0.000	0.022	0.020	**0.369**	0.019	0.007	

根据表 1,查找每一列(行)MIC 的最大值记为 $MIC_{\max}$.如果随机变量 X 和 Y 之间的 MIC 满足式(1),则认为在变量 X 和 Y 之间可能存在一条边,如表 1 中粗体标识.

$$\begin{cases}MIC(X,Y)\geq\alpha MIC_{\max}(X)\\MIC(X,Y)\geq\alpha MIC_{\max}(Y)\end{cases}\tag{1}$$

其中,α 为控制因子,一般取值为 0.9[12],以保证初始网络中包含标准网络中尽可能多的边.

根据表 1 依次在节点间添加有向边,其中设定边的方向为列节点指向行节点(如 A→T,也可设定行节点指向列节点),直到所有的列判断完毕得到非连通有向图.在添加有向边时有 3 条加边规则:

(1)为避免造成边的冗余,根据 MIC 值从大到小依次添加有向边时,若从该节点出发有其它边存在,则跳过该列继续判断下一列.如添加边 S→B 后,则不再判断边 S→L.

(2)在添加有向边时,若该边的反向边已经存在,则继续判断下一对满足式(1)的节点对.如从大到小依次添加 E→L 时,发现其反向边 L→E 已经存在,则继续判断 X→E.

(3)在添加有向边时,若构成环结构,则继续判断下一对满足式(1)的节点对.如若存在边 D→S 满足式(1),在添加该边时发现构成环结构 S→B,B→D,D→S,则跳过该边,继续判断下一个满足条件的节点对.

根据加边规则依次添加有向边构造有向图,如图 1(a)~图 1(h)所示.可以发现该方法不能保证得到的有向图一定为连通图,为此需要在连通分量之间添加有向边从而得到连通有向图.假设一个非连通图由 S 个连通分量组成,记为 $G=(G_1,G_2,\cdots,G_s)$,$G_i=\langle V_i,E_i\rangle$.若将非连通图修复成连通图,需要在 S 个连通分量中增加 $S-1$ 条边[31].其具体方法如下:假设有数据集 D,变量集 V_k 和 V_t,为使得变量集 V_k 和 V_t 相连通,则需要在两个变量集之间添加一条边.连通分量间的 $MIC_{\max}$ 应满足下列条件,$X_i\in V_k$,$X_j\in V_t$.

$$MIC_{\max}(V_k,V_t)=\max MIC(X_i,X_j)\tag{2}$$

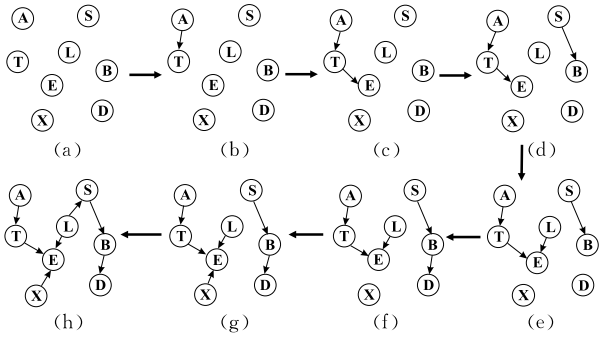


图 1 初始网络构建过程

计算 2 个联通分量任意节点之间的 MIC 值，并根据式(2)每次选择取值最大的 2 个联通分量，在 X_i 和 X_j 之间添加一条有向边并构成新的连通分量. 依此类推，直到将非连通图修复为连通图为止，如图 1(h)所示.

算法 1. 初始网络构建算法.

1. 表 1 每列中的最大值记为 $MIC_{\max}$，边集合 $E=\{\}$，n 为节点数
2. FOR col=1:n //遍历所有列
3. FOR row=1:n //遍历每一列中的所有行
4. IF $MIC(X_i,Y_j)\geq 0.9MIC_{\max}(Y)$ //查看是否满足式(1)
5. list=$sort(MIC_{X_i,X_j})$ //从大到小进行排序
6. FOR l=1:len(list) //遍历 list 列表
7. IF $(X_j,X_i)\in E$ or $isCycle(X_j,X_i)$
8. CONTINUE
9. ELSE
10. $Add(X_i,X_j)$ to E
11. $E\rightarrow dag$ //根据边集 E 构建初始网络
12. RETURN dag //返回初始网络

4.2 EFBNSL 网络结构学习算法

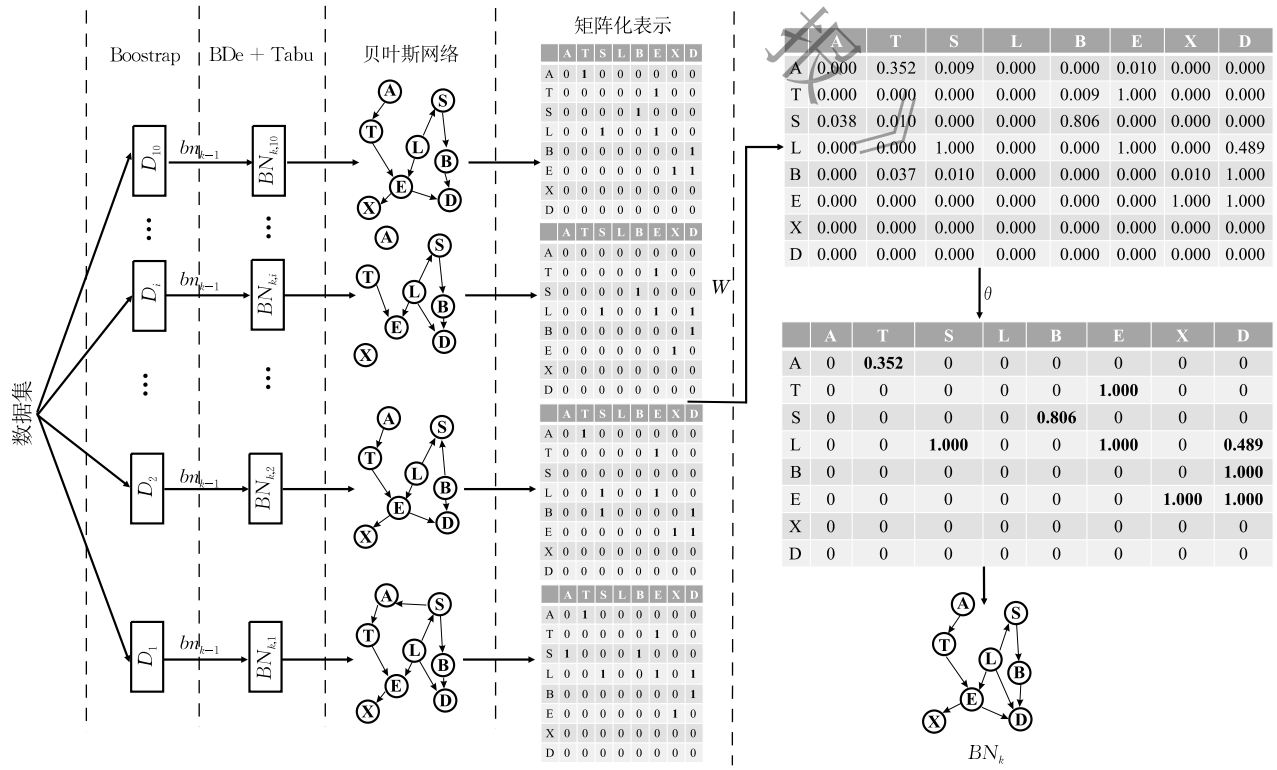
4.2.1 集成学习策略

本文采用集成学习思想，提出了集成策略函数 W 用于计算每条边的最终得分，其代表了该边的可信度，得分越高可信度越高. 集成策略函数主要为矩阵间的相关运算，因此最终得到的矩阵表示了各节点间边的得分情况，其中 W 定义如下：

$$W=\sum_{i=1}^{n}\sqrt{\frac{N_{\text{node}}}{N_{ki}}\boldsymbol{M}_{ki}}\odot\frac{\sum_{i=1}^{n}\boldsymbol{M}_{ki}}{n}\log(D) \tag{3}$$

式中，n 表示采样次数；N_{ki} 表示结合初始网络 bn_{k-1}，利用第 i 份训练样本学习到的边数；N_{node} 表示数据集的节点数；$\boldsymbol{M}_{ki}$ 表示结合初始网络 bn_{k-1}，利用第 i 份训练样本学习到的贝叶斯网络的矩阵化表示；D 表示样本量；$\odot$ 表示哈达玛积.

首先，利用 Bootstrap 采样方法对原数据进行采样. 设定每 10 次结构学习进行一次集成学习(即 $n=10$)，如图 2 所示. 对于每个训练样本，将初始网络 bn_{k-1}(k 轮迭代后)作为先验知识用于初始化



	A	T	S	L	B	E	X	D
A	0.000	0.352	0.009	0.000	0.000	0.010	0.000	0.000
T	0.000	0.000	0.000	0.000	0.009	1.000	0.000	0.000
S	0.038	0.010	0.000	0.000	0.806	0.000	0.000	0.000
L	0.000	0.000	1.000	0.000	0.000	1.000	0.000	0.489
B	0.000	0.037	0.010	0.000	0.000	0.000	0.010	1.000
E	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1.000
X	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
D	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

	A	T	S	L	B	E	X	D
A	0	**0.352**	0	0	0	0	0	0
T	0	0	0	0	0	**1.000**	0	0
S	0	0	0	0	**0.806**	0	0	0
L	0	0	**1.000**	0	0	**1.000**	0	**0.489**
B	0	0	0	0	0	0	0	**1.000**
E	0	0	0	0	0	0	**1.000**	**1.000**
X	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0

图 2 集成学习策略

Tabu 搜索算法,并结合 BDe 评分函数进行贝叶斯网络结构学习,并将学习到的贝叶斯网络矩阵化表示为 $\boldsymbol{M}_{ki}$. 其次,根据式(3)计算各边的得分矩阵 $\boldsymbol{W}_k$,并进行最大最小值归一化处理. 设定阈值 $\theta=0.35$,若 $\boldsymbol{W}_k[q,p]<\theta$,设置 $\boldsymbol{W}_k[q,p]=0$;否则 $\boldsymbol{W}_k[q,p]=1$,得到 01 矩阵 $\boldsymbol{W}_k$. 最后,对矩阵 $\boldsymbol{W}_k$ 进行遍历,若 $\boldsymbol{W}_k[q,p]=1$,则在空网络中添加有向边 $q\rightarrow p$,最终得到贝叶斯网络 BN_k. 若阈值 θ 设置过小,则贝叶斯网络中会出现大量冗余边;若 θ 设置过大,虽然冗余边会大大减少,但少边的情况会非常严重.

4.2.2　反馈策略

反馈策略是根据集成学习结果对初始网络进行更新(调节边的方向),来提高初始网络作为先验知识的有效性. C 为更新条件,其定义如下:通过判断 BN_k 与数据集的拟合程度来确定是否对初始网络 bn_{k-1} 进行更新,如图 3 所示. 其中设定 $score^*$ 表示当前最佳网络评分. 若 $score(BN_k,D)\geqslant score^*$,说明通过初始网络 bn_{k-1} 得到的贝叶斯网络 BN_k 较优,则不对初始网络进行更新;若 $score(BN_k,D)<score^*$,说明通过初始网络 bn_{k-1} 得到的网络 BN_k 较差,则执行 $update(\boldsymbol{M}_{k-1},\boldsymbol{W}_k)$ 对初始网络进行更新. 其中,$\boldsymbol{M}_{k-1}$ 表示初始网络 bn_{k-1} 的矩阵化表示,$\boldsymbol{W}_k$ 表示贝叶斯网络 BN_k 的矩阵化表示,如图 3 所示.

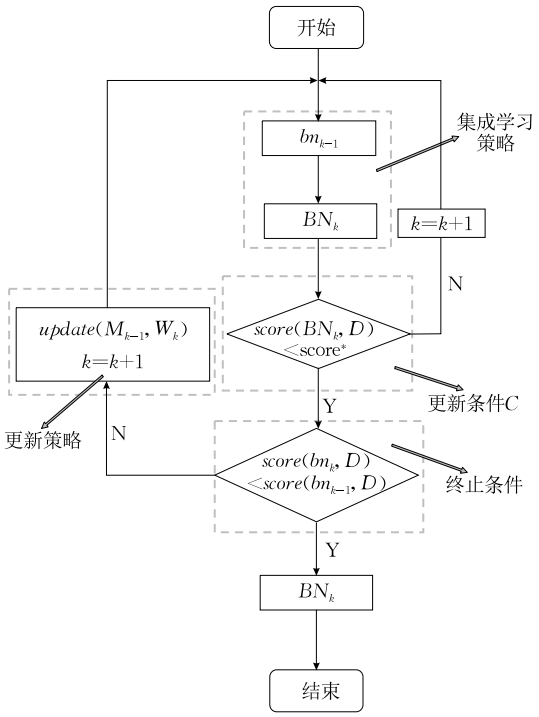


图 3　EFBNSL 算法流程图

4.2.3　更新策略

根据 $\boldsymbol{W}_k$ 对初始网络 bn_{k-1} 中边的方向进行调整. 具体做法:设初始网络 bn_{k-1} 的矩阵化表示为 $\boldsymbol{M}_{k-1}$,$\boldsymbol{W}_k$ 矩阵的转置为 $\boldsymbol{W}_k^{\mathrm{T}}$. 对 $\boldsymbol{M}_{k-1}$ 和 $\boldsymbol{W}_k^{\mathrm{T}}$ 的每个元素(两个矩阵都为 01 矩阵,即边存在为 1,否则为 0)进行判断,若 $\boldsymbol{M}_{k-1}[q,p]=\boldsymbol{W}_k^{\mathrm{T}}[q,p]$,则设 $\boldsymbol{M}_{k-1}[p,q]=1$,$\boldsymbol{M}_{k-1}[q,p]=0$,完成更新并得到初始网络 bn_k 进入下一轮迭代. 直到满足终止条件 $score(bn_k,D)<score(bn_{k-1},D)$ 时,返回当前贝叶斯网络,即得到最优贝叶斯网络(如图 3 所示).

算法 2.　EFBNSL 算法流程.

输入:初始网络 bn_{k-1},更新频率 $n=10$,条件阈值 θ,初始网络邻接矩阵 $\boldsymbol{M}_{k-1}$,结果矩阵 $\boldsymbol{W}_k$,数据样本量 D,节点个数 m,$\boldsymbol{M}_{ki}$,$\boldsymbol{M}_{ks}$,$\boldsymbol{M}_{kw}$ 初始化为空矩阵

输出:贝叶斯网络 DAG

```
1.  DO
2.    k=1
3.    FOR i=1:n DO
4.      BN_ki=tabu(bn_{k-1},BDe,D_i)
5.      BN_ki→M_ki
6.      M_ks=M_kx+M_ks   //统计每条边学习到的次数
7.      M_kx=sqrt(m/N_ki * M_kx)   //为每条边赋予权重
8.      M_kw=M_kx+M_kw   //计算每条边总的权重值
9.    W_k=M_kw⊙(M_ks/n)log(D) //根据式(3)进行计算
10.   W_k=normalize(W_k) //对各边权重进行归一化表示
11.   FOR q=1:m
12.     FOR p=1:m
13.       IF W_k[q,p]<θ   //M_k表示集成算法所得网络矩阵化表示
14.         W_k[q,p]=0   //0 表示节点之间不存在边
15.       ELSE
16.         W_k[q,p]=1   //1 表示节点之间存在边
17.   W_k→BN_k //将邻接矩阵W_k转换为贝叶斯网络BN_k
18.     IF score(BN_k,D)<score*   //score* 表示当前最优评分
19.       M_k=update(M_{k-1},W_k)
20.     ELSE
21.       score(BN_k,D)→score*
22.     k=k+1   //进入下一轮迭代
23. WHILE score(bn_k,D)<score(bn_{k-1},D)
                    //直到满足终止条件
24.     RETURN BN_k   //返回最优贝叶斯网络 BN_k
```

4.2.4　时间和空间复杂度分析

在本节中,我们对 EFBNSL 算法的时间复杂度和空间复杂度进行分析. 该算法整体分为以下几个部

分,接下来对每个部分进行时间复杂度分析.假设 n 为网络节点个数,m 为样本量.在构建初始网络时,需要计算节点间的 MIC 值,由于 MIC 具有对称性,因此只需要计算 $n(n-1)/2$ 次 MIC.计算节点间 MIC 值的时间复杂度是多项式级的,用 $mic(m,r)$ 表示[12,33],其中 r 为任意变量可能取值的最大值,因此计算所有节点间 MIC 值的时间复杂度为 $O(n^2 \cdot mic(m,r))$.对所有节点的 MIC 值进行排序(快速排序)其时间复杂度为 $O(n\log n)$.在根据 MIC 值构建初始网络时(随机定向),其时间复杂度为 $O(n^2)$,因此该部分的时间复杂度为 $O(n^3)$.在连接所有的连通分量时,需要 $O(n^2)$ 去找到所有的连通分量,这里最多需要 $n/2-1$ 次操作去连接各连通分量,因此其时间复杂度为 $O(n^3)$.所以构建最终的初始网络部分的时间复杂度为 $O(n^2 \cdot mic(m,r)+n^3)$,它的最坏时间复杂度是关于节点数的多项式和关于样本量的线性表达[12].在学习贝叶斯网络结构时,禁忌搜索算法的时间复杂度为 $O(n^2)$,计算 BDe 分值所需要的时间是 $O(mn)$[10,20],因此其时间复杂度为 $O(n^3)$.两个 $n\times n$ 的矩阵作哈达玛积、矩阵相加、矩阵的归一化处理、将矩阵转换为有向图、将有向图转换为矩阵操作,其时间复杂度都为 $O(n^2)$.假设 MAX_ITER 为最大迭代次数,因此学习贝叶斯网络结构部分的时间复杂度为 $O(MAX_ITER \cdot n^3)$.因此该算法总的时间复杂度为 $O(n^2 \cdot mic(m,r))+O(MAX_ITER \cdot n^3)$.空间复杂度分析,禁忌搜索算法由局部搜索算法发展而来,而局部搜索算法往往能够以常数的空间复杂度在一定空间中找到合理解.而对于矩阵的相关操作其空间复杂度为 $O(n^2)$,因此该算法总的空间复杂度为 $O(n^2)$,因此该算法的空间复杂度是关于节点数的多项式表达.

5 实验与结果分析

5.1 实验准备

本文选用 R 语言①中的 bnlearn 工具包②,通过 R 语言编程进行贝叶斯网络结构学习.bnlearn 包主要用于贝叶斯网络结构学习、参数估计等[38].本文使用的是 4.3 版本.实验选用 7 个大小不同的标准网络进行对比实验,数据集获取首先从 Bayesian Network Repository③ 下载 RDA 文件并加载 BN,然后使用 rbn 函数生成相应大小的样本,具体情况如表 2 所示.实验的运行环境:操作系统为 Windows10,64 bit,CPU 为 2.5 GHz,内存为 16 GB.

表 2 标准数据集介绍

数据集	样本量	节点	边	最大父节点数
ASIA	10000	8	8	2
ALARM	20000	37	46	4
INSURANCE	20000	27	52	3
CHILD	20000	20	25	2
BARLEY	20000	48	84	4
HEPARII	20000	70	123	6
WIN95PTS	20000	76	112	7

5.2 评价指标

为了验证算法的有效性,本文选用 $F1$ 值($F1$-score,式(6)所示)和汉明距离(Hamming Distance,式(7)所示)对生成的贝叶斯网络进行评价.其中,$recall$ 表示召回率,$precision$ 表示精准率,TP 表示在标准网络和当前网络中均存在的边数,即学习到的正确边数;FP 表示存在于当前网络中而不在标准网络中出现的边的个数;FN 表示存在于标准网络中而不在当前网络中出现的边的个数.HD 值越小表示学习到的贝叶斯网络越接近真实网络.

$$recall=\frac{TP}{(TP+FN)} \tag{4}$$

$$precision=\frac{TP}{(TP+FP)} \tag{5}$$

$$F_1=\frac{2\times recall\times precision}{recall+precision} \tag{6}$$

$$HD=FP+FN \tag{7}$$

5.3 结果分析

实验中对比了 3 种不同大小的标准网络,包括“小型”网络 ASIA 数据集,“中型”网络 ALARM、INSURANCE、CHILD、BARLEY 数据集和“大型”网络 HEPARII、WIN95PTS 数据集,均按 1K,2K,5K 的样本量进行采样.实验与其它结构学习算法如:PC[39]、GS、IAMB[40]、HC[41]、MMHC[11]、TS、DiG-SiRGA[42] 和 ESL-GA[28] 在 $F1$ 值、HD、正确边数(TP)这 3 个指标上进行对比分析,结果为 10 次学习后的平均值和相应的标准差(表 4 括号中表示).实验中设定权重阈值 $\theta=0.35$,该值的选择对最终结果的影响并不敏感,可以在一定范围内取值.我们在 ALARM 数据上当样本量为 1K 时进行了实验,如表 3 所示.

① https://www.r-project.org
② http://www.bnlearn.com
③ http://www.bnlearn.com/bnrepository/

《计算机学报》

表 3　不同 θ 取值对各指标的影响

θ	$F1$	HD	TP
0.05	0.768	21.8	36.0
0.10	0.785	19.6	35.7
0.20	0.774	20.1	34.5
0.35	**0.776**	**19.1**	**33.3**
0.45	0.701	24.1	28.4
0.60	0.702	22.4	26.5
0.80	0.669	23.4	23.7
0.90	0.566	28.2	18.4

通过对表 3 分析可知,当权重阈值设置过小时,会有更多的边被选中并加入到网络中.此时可能会有更多的正确边,但冗余边的情况也会很严重,即 TP 和 HD 都会相对较高.当权重阈值选择过大时,使得更少的边被选中并加入到网络中.导致很多可能正确的边没有被加入网络,造成正确边的数量较少,同时缺边的情况也会非常严重,因此 TP 会相对较小而 HD 会相对较高.综合分析,合理选择阈值大小是准确进行网络结构学习的前提.在保证有更多正确边的同时,也使得多边和缺边的情况较轻.

从表 4 中可以看出,本文提出的算法相较于其它结构学习算法均取得了较好的学习效果.对于 ASIA 网络,当样本量较少时(1K),EFBNSL 算法没有表现出较大的优势,当样本量>1K 时,EFBNSL 算法学习效果较好,其中 HC 和 TS 算法也取得了相对较好的结果.通过观察发现 HC 和 TS 算法学习到正确边的数量大于等于 EFBNSL 算法学习到正确边的数量,而 HD 值高于 EFBNSL 算法的 HD 值.这说明 EFBNSL 算法可以有效减少网络中可能存在的无效边的数量.对于节点较多、网络结构相对复杂的 ALARM 网络,EFBNSL 算法在学习效果上有较明显的优势.对于"大型"网络 HEPARII,EFBNSL 算法也表现较好.其中在样本量为 2K 时,HC 和 TS 算法类似于在 ASIA 网络中的表现,同样表现出较高的正确边数和 HD 值,说明集成学习算法相较于单一学习算法具有一定的优势.

表 4　不同数据集下 EFBNSL 算法和其它算法性能对比

方法	ASIA 1K			ASIA 2K			ASIA 5K		
	$F1$	HD	TP	$F1$	HD	TP	$F1$	HD	TP
PC	0.313(0.010)	8.8(0.4)	2.0(0.0)	0.320(0.038)	8.9(0.5)	2.1(0.3)	0.310(0.070)	9.3(1.0)	2.1(0.5)
GS	0.349(0.092)	7.9(1.3)	2.1(0.5)	0.349(0.092)	7.9(1.3)	2.1(0.5)	0.441(0.083)	6.9(1.2)	2.7(0.5)
IMBA	0.360(0.061)	7.8(0.7)	2.2(0.7)	0.463(0.066)	6.5(0.8)	2.8(0.4)	0.429(0.079)	6.9(0.9)	2.6(0.9)
HC	**0.837**(0.094)	2.6(1.6)	**6.6**(0.7)	0.897(0.065)	1.6(1.0)	**7.0**(0.6)	0.898(0.056)	1.6(0.9)	**7.0**(0.4)
MMHC	0.738(0.047)	3.3(0.5)	4.7(0.5)	0.772(0.033)	3.0(0.4)	5.1(0.3)	0.761(0.062)	3.1(0.7)	5.0(0.6)
TS	0.813(0.086)	3.0(1.5)	6.4(0.5)	0.844(0.060)	2.4(0.9)	6.5(0.5)	0.876(0.096)	2.0(1.5)	**7.0**(0.8)
DiG-SiRGA	0.795(0.045)	3.1(1.0)	5.1(0.8)	0.820(0.054)	2.9(1.1)	6.6(0.8)	0.832(0.055)	2.6(0.6)	6.7(0.5)
ESL-GA	0.817(0.060)	2.9(0.9)	6.5(0.7)	0.804(0.057)	3.0(0.9)	6.1(0.3)	0.822(0.062)	2.9(1.1)	6.6(0.5)
EFBNSL	0.835(0.058)	**2.4**(0.8)	**6.6**(0.5)	**0.908**(0.043)	**1.4**(0.7)	6.9(0.3)	**0.921**(0.046)	**1.2**(0.7)	**7.0**(0.4)

方法	ALARM 1K			ALARM 2K			ALARM 5K		
	$F1$	HD	TP	$F1$	HD	TP	$F1$	HD	TP
PC	0.270(0.070)	50.5(3.9)	9.4(2.7)	0.398(0.115)	46.3(8.8)	15.3(4.5)	0.540(0.049)	36.5(3.4)	21.5(2.2)
GS	0.057(0.039)	59.6(2.5)	1.8(1.2)	0.078(0.022)	59.7(2.9)	2.5(0.7)	0.154(0.039)	57.2(2.4)	5.2(1.3)
IMBA	0.334(0.084)	48.5(5.9)	12.2(3.1)	0.338(0.077)	49.9(6.2)	12.7(2.8)	0.409(0.081)	45.7(5.8)	15.9(3.4)
HC	0.564(0.081)	43.8(9.3)	28.1(3.6)	0.592(0.061)	40.4(6.8)	29.2(2.6)	0.615(0.063)	38.5(7.1)	30.6(2.6)
MMHC	0.568(0.045)	31.6(2.9)	20.8(2.0)	0.633(0.024)	27.8(1.7)	24.0(1.1)	0.668(0.027)	26.2(2.2)	26.4(1.1)
TS	0.555(0.043)	45.7(5.4)	28.4(2.0)	0.627(0.068)	37.0(6.7)	31.2(3.7)	0.676(0.042)	31.8(4.4)	33.2(1.9)
DiG-SiRGA	0.653(0.054)	36.7(3.4)	25.7(2.0)	0.683(0.034)	32.2(2.2)	28.4(1.2)	0.690(0.064)	34.5(2.5)	30.0(1.4)
ESL-GA	0.675(0.035)	35.0(2.4)	29.5(0.9)	0.680(0.021)	27.2(1.5)	29.5(0.8)	0.688(0.033)	23.2(1.3)	32.4(0.9)
EFBNSL	**0.787**(0.066)	**19.0**(6.0)	**35.0**(2.8)	**0.839**(0.017)	**14.5**(1.5)	**37.9**(0.9)	**0.851**(0.021)	**13.6**(2.0)	**38.8**(1.0)

方法	HEPARII 1K			HEPARII 2K			HEPARII 5K		
	$F1$	HD	TP	$F1$	HD	TP	$F1$	HD	TP
PC	0.122(0.037)	141.0(5.6)	9.8(3.0)	0.174(0.043)	137.5(6.4)	14.5(3.7)	0.214(0.044)	138.7(7.2)	18.9(4.0)
GS	0.147(0.027)	128.5(4.5)	11.1(2.0)	0.167(0.036)	127.2(5.0)	12.8(2.8)	0.209(0.032)	122.1(4.7)	16.1(2.6)
IMBA	0.148(0.041)	137.8(6.6)	12.0(3.3)	0.218(0.052)	132.1(7.5)	18.5(4.6)	0.283(0.072)	127.4(12.7)	25.2(6.5)
HC	0.486(0.017)	105.0(5.6)	49.5(1.7)	0.571(0.013)	87.7(2.2)	**58.4**(2.1)	0.652(0.018)	74.6(4.7)	69.9(1.6)
MMHC	0.389(0.013)	98.2(1.5)	31.3(1.3)	0.435(0.023)	95.5(4.0)	36.7(2.1)	0.504(0.021)	88.9(4.1)	45.1(1.9)
TS	0.472(0.030)	108.6(6.9)	48.5(3.2)	0.561(0.029)	91.1(7.5)	**58.2**(2.6)	0.644(0.029)	76.9(6.5)	69.6(3.0)
DiG-SiRGA	0.456(0.010)	87.6(1.6)	36.7(0.8)	0.461(0.007)	86.8(1.1)	37.1(0.5)	0.518(0.006)	81.0(1.0)	43.5(0.5)
ESL-GA	0.423(0.012)	91.2(1.8)	33.4(0.9)	0.433(0.005)	89.6(0.8)	34.2(0.4)	0.487(0.004)	84.2(0.6)	39.9(0.3)
EFBNSL	**0.520**(0.019)	**88.4**(4.7)	**47.9**(1.6)	**0.586**(0.013)	**80.8**(2.6)	57.3(1.6)	**0.674**(0.014)	**68.0**(3.7)	**70.2**(1.0)

对不同样本量下的学习结果分析发现,相较于少样本量(1K),在样本量较大(2K,5K)时均能取得更好的学习效果.对于"小型"网络 ASIA,在样本量较少时也能取得不错的学习效果.因为该网络较小,

在样本量为 1K 时便能够较充分地进行结构学习. 对于网络结构较复杂的 ALARM 和 HEPARII 数据集,当样本量较少时学习并不充分,随着样本量的增加会得到更充分的学习,取得更好的学习结果.

为了验证 EFBNSL 算法的泛化能力的强弱,我们在更多标准数据集上进行了实验,如图 4 所示. 对图 4 进行分析发现,EFBNSL 算法在各数据集上均有较好的表现,说明该算法具有普适性. 将初始网络作为先验知识用于结构学习,能够有效压缩搜索空间,提高算法的搜索效率. 而且初始网络的好坏会直接影响最终的学习结果,特别对于稀疏网络. 在学习过程中,通过不断调整优化初始网络使得初始网络不断接近真实网络,因此生成的贝叶斯网络也会越接近真实网络,最终的学习结果也会趋于最优. 但是初始网络是通过对变量间的*MIC*值进行检测得到的,与标准网络相比还存在一定的误差(如多边、反边),因此初始网络可能会对网络结构的学习产生负面影响,这也是在某些数据集上的学习效果为次优的原因.

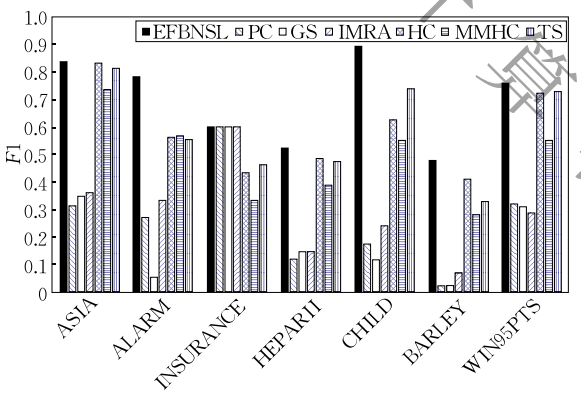


图 4 各算法在不同数据集上的效果对比

通过上述分析发现,本文提出的 EFBNSL 算法在进行贝叶斯网络结构学习时,(1)对于节点较少的贝叶斯网络,在样本量较少时可以对网络进行较全面的描述,节点间的关系可以较准确的表示;(2)对于多节点的大型网络,集成学习算法相较于单一的学习算法,结构学习效果均有不错的表现,但要充分考虑初始网络可能带来的负面影响;(3)基于集成学习思想的 EFBNSL 算法在不同大小的数据集上均取得相对不错的结果,能够降低结构学习中存在的多边、少边、反边的情况,增加了算法的稳定性和泛化能力.

通过对 EFBNSL 算法进行分析发现,利用学习结果不断反馈更新初始网络,使得初始网络变得更加有效(有更少的反向边). 而利用更有效的初始网络会学习到更优的贝叶斯网络. 因此在迭代过程中,学习到的网络会不断接近于真实网络,从而最终得到最优的网络学习结果. 为了展示在训练过程中贝叶斯网络的学习情况,在不同数据集下对训练过程中各指标的变化情况进行了可视化显示,如图 5 所示. 对各图分析发现,随着迭代次数的增加,学习结果会逐渐变好,即 $F1$ 值和 TP 值不断增加而 HD 值不断减小. 但由于在计算贝叶斯网络得分时,马尔可夫等价类的得分相等,所以在迭代过程中,当初始网络获得最高得分而退出循环时,得到的贝叶斯网络可能存在马尔可夫等价类,而出现如图 5(b)所示的情况. 但通过集成学习和利用初始网络作为先验知识用于贝叶斯网络结构学习,相对于单一的结构学习算法(如基于评分搜索的方法)在很大程度上会改善此问题,从而提高贝叶斯网络结构学习的有效性.

6 结 论

本文结合最大信息系数 *MIC* 和集成学习思想到贝叶斯网络结构学习中,提出了 EFBNSL 算法. 通过计算节点间的 *MIC* 值,确定初始网络结构,进而将 Tabu 搜索算法和 BDe 评分函数相结合采用集成学习和反馈更新策略进行贝叶斯网络结构学习,提高了算法的有效性. 通过在 7 种标准网络上进行实验,并与其它的结构学习算法进行对比,验证了 EFBNSL 算法在结构学习上的有效性和普适性. 对不同的结构学习算法进行集成,充分利用不同结构学习算法的优势进行贝叶斯网络结构学习将是下一步的研究重点.

参 考 文 献

[1] Liang C, Ghazel M, Cazier O, El-Koursi E M. Risk analysis on level crossings using a causal Bayesian network based approach. Transportation Research Procedia, 2017, 25(C): 2172-2186

[2] Liu Yin-Hua, Sun Jin. Application of Bayesian networks for diagnostics in the assembly process by considering small measurement data sets. International Journal of Advanced Manufacturing Technology, 2013, 65(9-12): 1229-1237

[3] Nistal-Nuno B. Tutorial of the probabilistic methods Bayesian networks and influence diagrams applied to medicine. Journal of Evidence-Based Medicine, 2018, 11(2): 112-124

[4] Drury B, Valverde-Rebaza J, Moura M F, de Andrade Lopes A. A survey of the applications of Bayesian networks in agriculture. Engineering Applications of Artificial Intelligence, 2017, 65: 29-42

[5] Chickering D M, Geiger D, Heckerman D. Learning Bayesian

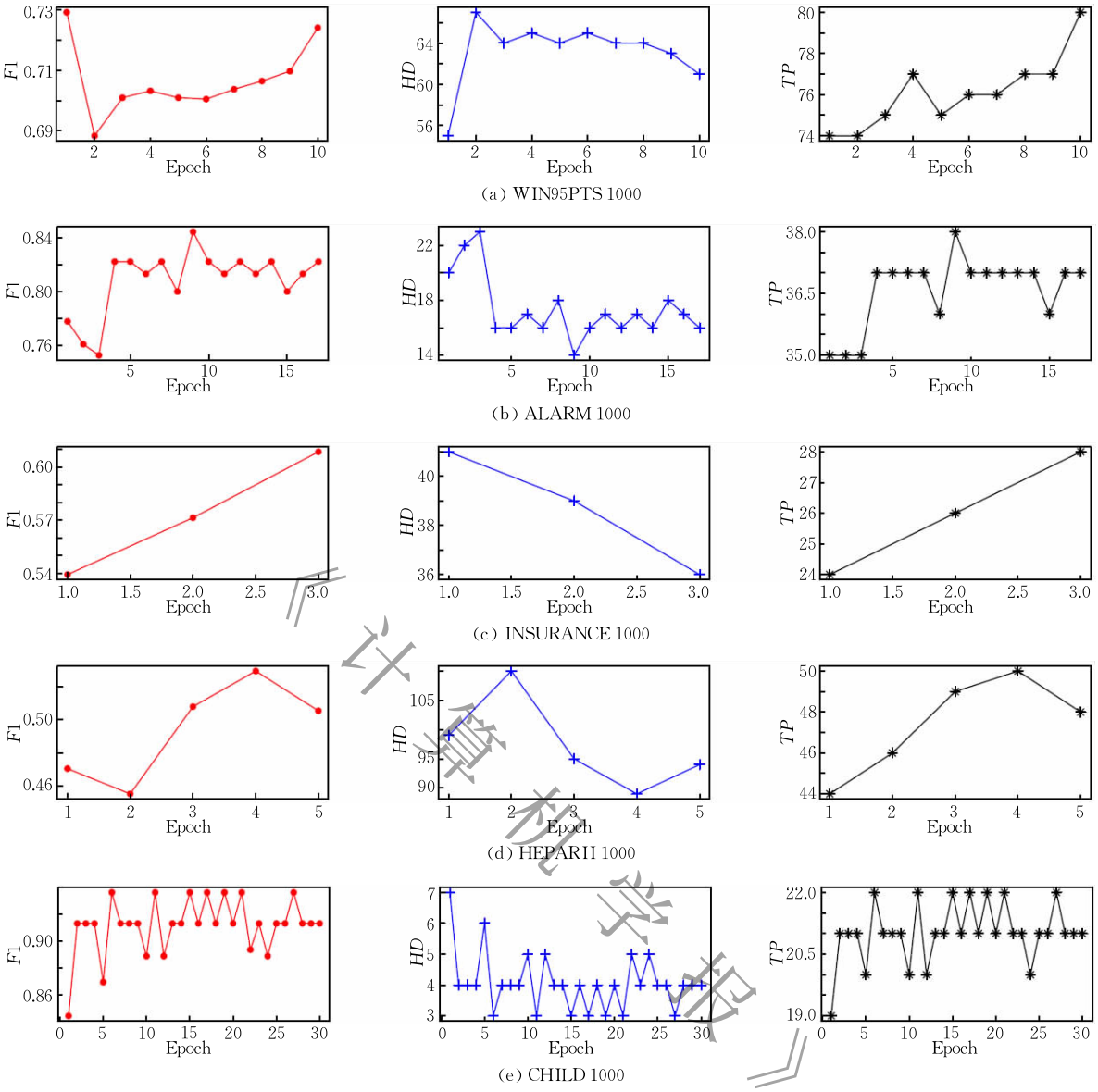


图 5　不同数据集下 EFBNSL 算法迭代学习情况

networks is NP-complete. Networks, 1996, 112(2): 121-130

[6] Xie Xian-Chao, Geng Zhi, Zhao Qiang. Decomposition of structural learning about directed acyclic graphs. Artificial Intelligence, 2006, 170(4-5): 422-439

[7] Schwarz G. Estimating the dimension of a model. 1978, 6(2): 461-464

[8] Lam W, Bacchus F. Learning Bayesian belief networks: An approach based on the MDL principle. Computational Intelligence, 1994, 10(3): 269-293

[9] Heckerman D, Geiger D, Chickering D M. Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning, 1995, 20(3): 197-243

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

[11] Tsamardinos I, Brown L E, Aliferis C F. The max-min hill-climbing Bayesian network structure learning algorithm. Machine Learning, 2006, 65(1): 31-78

[12] Zhang Ying-Hua, Zhang Wen-Sheng, Xie Yuan. Improved heuristic equivalent search algorithm based on maximal information coefficient for Bayesian network structure learning. Neurocomputing, 2013, 117: 186-195

[13] Cai Qing-Song, Chen Xi-Hou. Bayesian network structure merging algorithm based on scoring function. Computer Engineering and Applications, 2019, 55(11): 147-152(in Chinese)
(蔡青松, 陈希厚. 基于评分函数的贝叶斯网络结构融合算法. 计算机工程与应用, 2019, 55(11): 147-152)

[14] Spirtes P, Meek C. Learning Bayesian networks with discrete variables from data//Proceedings of the 1st International Conference on Knowledge Discovery & Data Mining. Montreal, Canada, 1995: 294-299

[15] Spirtes P, Glymour C, Scheines R. Causation, Prediction, and Search. 2nd Edition. Cambridge, USA: MIT Press,

2001

[16] Cheng J, Greiner R, Kelly J, et al. Learning Bayesian networks from data: An information-theory based approach. Artificial Intelligence, 2001, 137(1-2): 43-90

[17] Hong Ying-Han. Fast causal network skeleton learning algorithm. Journal of Nanjing University of Science and Technology, 2016, 40(3): 315-321(in Chinese)
(洪英汉. 一种快速因果网络骨架学习算法. 南京理工大学学报, 2016, 40(3): 315-321)

[18] Constantinou A C. Evaluating structure learning algorithms with a balanced scoring function. arXiv preprint arXiv: 1905.12666, 2019

[19] Song K, Kim D W. An efficient node ordering method using the conditional frequency for the K2 algorithm. Pattern Recognition Letters, 2014, 40(4): 80-87

[20] Chen Xue-Wen, Anantha G, Lin Xiao-Tong. 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

[21] Liu Bin, Wang Hai-Yu, Sun Mei-Ting, et al. Learning Bayesian network structure from node ordering searching optimal. Journal of Electronics & Information Technology, 2018, 40(5): 1234-1241(in Chinese)
(刘彬, 王海羽, 孙美婷等. 一种通过节点序寻优进行贝叶斯网络结构学习的算法. 电子与信息学报, 2018, 40(5): 1234-1241)

[22] Fan Xian-Nian, Yuan Chang-He. An improved lower bound for Bayesian network structure learning//Proceedings of the 29th Conference on Artificial Intelligence. Austin, USA, 2015: 3526-3532

[23] Gheisari S, Meybodi M R, Dehghan M, Ebadzadeh M M. Bayesian network structure training based on a game of learning automata. International Journal of Machine Learning and Cybernetics, 2017, 8(4): 1093-1105

[24] Liu Hao-Ran, Zhang Li-Yue, Fan Rui-Xing, et al. Bayesian network structure learning based on improved whale optimization strategy. Journal of Electronics & Information Technology, 2019, 41(6): 1434-1441(in Chinese)
(刘浩然, 张力悦, 范瑞星等. 基于改进鲸鱼优化策略的贝叶斯网络结构学习算法. 电子与信息学报, 2019, 41(6): 1434-1441)

[25] Askari M B A, Ahsaee M G. Bayesian network structure learning based on cuckoo search algorithm//Proceedings of the 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems(CFIS). Kerman, Iran, 2018: 127-130

[26] Ji Jun-Zhong, Wei Hong-Kai, Liu Chun-Nian. An artificial bee colony algorithm for learning Bayesian networks. Soft Computing, 2013, 17(6): 983-994

[27] Ji Jun-Zhong, Zhang Hong-Xun, Hu Ren-Bing, Liu Chun-Nian. Learning Bayesian network structure based on ant colony optimization algorithms. Journal of Beijing University of Technology, 2011, 37(6): 933-939, 954

[28] Contaldi C, Vafaee F, Nelson P C. Bayesian network hybrid learning using an elite-guided genetic algorithm. Artificial Intelligence Review, 2018, (293): 1-28

[29] Vafaee F. Learning the structure of large-scale Bayesian networks using genetic algorithm//Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. Vancouver, Canada, 2014: 855-862

[30] Cai Zhi-Qiang, Si Shu-Bin, Sun Shu-Dong, Dui Hong-Yan. Learning Bayesian network structure with immune algorithm. Journal of Systems Engineering and Electronics, 2015, 26(2): 282-291

[31] Zeng Qian-Qian, Zeng An, Pan Dan, et al. Bayesian network structure learning algorithm based on maximal information coefficient. Computer Engineering, 2017, 43(8): 225-230(in Chinese)
(曾千千, 曾安, 潘丹等. 基于最大信息系数的贝叶斯网络结构学习算法. 计算机工程, 2017, 43(8): 225-230)

[32] Pearl J. Fusion, propagation, and structuring in belief networks. Artificial Intelligence, 1986, 29(3): 241-288

[33] Reshef D N, Reshef Y A, Finucane H K, et al. Detecting novel associations in large data sets, Science, 2011, 334(6062): 1518-1524

[34] Zhou Zhi-Hua, Wu Jian-Xin, Tang Wei. Ensembling neural networks: Many could be better than all. Artificial Intelligence, 2002, 137(1-2): 239-263

[35] Zhang Chun-Xia, Zhang Jiang-She. A survey of selective ensemble learning algorithms. Chinese Journal of Computers, 2011, 34(8): 1399-1410(in Chinese)
(张春霞, 张讲社. 选择性集成学习算法综述. 计算机学报, 2011, 34(8): 1399-1410)

[36] Hansen L K, Salamon P. Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12(10): 993-1001

[37] Breiman L. Bagging predicators. Machine Learning, 1996, 24(2): 123-140

[38] Scutari M. Learning Bayesian networks with the bnlearn R package. Journal of Statistical Software, 2010, 35(3): 1-22

[39] Colombo D, Maathuis M H. Order-independent constraint-based causal structure learning. Journal of Machine Learning Research, 2014, 15: 3921-3962

[40] Tsamardinos I, Aliferis C F, Statnikov A R. Algorithms for large scale Markov blanket discovery//Proceedings of the 16th International Florida Artificial Intelligence Research Society Conference. Augustine, USA. 2003: 376-381

[41] Buntine W L. Operations for learning with graphical models. Journal of Artificial Intelligence Research, 1994, 2(1): 159-225

[42] Contaldi C, Vafaee F, Nelson P C. The role of crossover operator in Bayesian network structure learning performance: A comprehensive comparative study and new insights//Proceedings of the Genetic and Evolutionary Computation Conference(GECCO). Berlin, Germany, 2017: 769-776

《计算机学报》

WANG Shou-Hui, Ph. D. candidate. His research interests include bayesian network and knowledge base question answering (KBQA).

QIN Biao, Ph. D., associate professor. His research interests include bayesian network and data mining.

Background

Bayesian Network (BN) is a classical tool of representation for uncertain knowledge and reasoning in artificial intelligence, and has widely used in data mining, medical diagnosis, pattern recognition, industrial control and other fields. Bayesian network learning mainly includes structural learning, parameter learning and inference, in which structural learning is the key and difficulty of Bayesian network research.

The goal of Bayesian Network Structure Learning (BNSL) is to learn a network structure that has the highest fit to the data set, but it has proven to be a NP-Hard problem. At present, BNSL methods can be classified into three categories: (1) constraint-based structural learning; (2) score-based structural learning; (3) hybrid-based structural learning. The core idea of these algorithms mainly includes the following aspects. First, to improve the efficiency of the search algorithm, and avoid the search algorithm fall into the local optimum. Second, to enhance the score function, which is used to measure the fitting degree of Bayesian network and observation dataset. By improving the score function, the algorithm is no longer restricted in one aspect (e. g. the K2 algorithm needs the order of the nodes as the prior knowledge). The accuracy of the algorithm can be improved and the over-fitting risk of the algorithm can be reduced. However, there is always a certain deficiency in a single algorithm, and ensemble learning which combines single models can reduce the randomness of learning results and improves the effectiveness and the generalization ability of the results. In this paper, we use Maximal Information Coefficient (MIC) to determine whether there is an edge between the random nodes. And then construct the initial network. Then we use the initial network to initialize the Tabu algorithm and combine the BDe score function for BNSL. To ensure the effectiveness and the generalization ability, we exploit ensemble learning to reduce the number of invalid edges (e. g. multi-edge, less-edge, and reverse-edge) in the network and employ the feedback strategy to update the initial network. Using the ensemble learning and the feedback strategy in an iterative way, our Ensemble-Feedback Bayesian Network Structure Learning (EFBNSL) algorithm can output an optimal network structure. Finally, we perform experiments using seven benchmarks with different sizes. The result shows that our EFBNSL algorithm outperforms the other algorithms and has better generalization ability. This research method can provide a reference for BNSL.

The main contributions of our work include: (1) we propose a method to evaluate the edge score, and use the ensemble learning to reduce the number of invalid edges that may exist in the Bayesian network; (2) we propose a novel structural learning algorithm, which employs the feedback strategy to adjust the edge directions of the initial network, and obtains the optimal network by an iterative optimization; (3) we use the edge-adding strategy to determine the directed edges and obtain the initial network. It is easier than the previous method, which first obtains the undirected edges, then eliminates the loops, and finally decides their directions.

This work is supported by the National Natural Science Foundation of China (No. 61772534).

《计算机学报》