

# 贝叶斯网络结构加速学习算法

SEIN Minn 傅顺开

(华侨大学计算机科学与技术学院 厦门 361021)

**摘要** 结构学习是应用贝叶斯网络(BN)的基础。提出一种新的基于约束的学习类算法 APC(Accelerated PC),它基于一系列局部结构的推导获得 BN。APC 不但继承了经典的 PC(Peter & Clark)算法优先执行低阶条件独立(CI)测试的优点,而且能够从已执行的 CI 测试中推导相关拓扑信息,并利用其来挑选并优先执行更可能  $d$ -分割节点  $X$  和  $Y$  的候选 CI 测试。该策略可有效避免在搜索过程中执行无效的 CI 测试,例如 APC 算法在实验中较 PC 算法节省高达 50% 的计算量,同时实现了质量相同的学习效果。

**关键词** 贝叶斯网络,结构学习,基于约束的学习,条件独立性测试

**中图分类号** TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.2.055

## Accelerating Structure Learning of Bayesian Network

SEIN Minn FU Shun-kai

(College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China)

**Abstract** Structure learning is the basis for the application of Bayesian networks (BN). A novel algorithm called APC was proposed to recovery the whole structure via sequential induction of local structures. APC inherits the most feature of PC algorithm, i. e. effectively avoiding high-dimensional conditional independence (CI) tests. Besides, it constructs and sorts candidate sets which possibly  $d$ -separate any pair of nodes,  $X$  and  $Y$ , based on information implied in early conducted CI tests and known features of BN topology. Then, CI tests involving highly ranked candidate set are performed with priority. This strategy is expected to avoid fruitless CI tests, and up to 50% saving is observed on APC over PC in our experimental study.

**Keywords** Bayesian network, Structure learning, Constraint-based learning, Conditional independence test

## 1 引言

贝叶斯网络(Bayesian Network, BN)属于概率图模型,可简洁有效地表示变量间的相互关系,为不确定性推理提供强有力的工具。目前, BN 已在多个领域的决策和诊断中获得成功应用,例如反应堆补水<sup>[1]</sup>、无人机控制<sup>[2]</sup>和医疗诊断<sup>[3]</sup>。在利用 BN 进行不确定推理之前,需要先根据领域知识构造出相应的模型。过去,模型的构造常由领域专家利用专业知识进行人工构造,但是该过程耗时且实现复杂。因此,从样本中自动学习 BN 已成为主要的构造途径<sup>[4]</sup>。

BN 包含结构和参数两部分信息,故相应的学习任务也分为参数和结构学习。参数学习假设结构已知,而后从样本数据中学习每个变量的(条件)概率分布。概率分布的形式一般已预先指定,如多项式分布、高斯分布和泊松分布等,只需利用一定的策略估计这些分布的参数。

BN 的结构学习是贝叶斯网络学习的关键任务,此时 BN 的参数与结构均未知,需要找到与样本数据匹配度最好的网络结构。结构学习的精度要求取决于不同的学习目标,知识

发现、密度估计和特定预测任务。知识发现旨在通过研究所学结构的节点依赖性,发现相关变量的依赖关系。这种依赖关系由结构中的边反映出来,所以学习的结构越精确越好,理想情况是恢复真实结构。密度估计,即估计真实分布的统计模型,理想目标是学习所得模型  $M$  对应的关联分布  $\rho^*$  和生成数据的分布  $\rho$  一致。特定预测任务,例如分类,即寻找能够最好地“回答”诸如  $P(Y|X)$  查询的模型,而密度估计是寻找  $P(Y, X)$ 。这 3 类学习目标中,知识发现难度最大,因为它的终极目标是恢复完整的网络结构;学习输出中哪怕缺失的仅是一条关于弱依赖关系的边,也被认为是错误的。而在密度估计中,这样的缺失因为对整体目标的影响不大而能被接受。

结构学习算法可分为 3 类:基于约束的(Constraint Based, CB)学习算法、基于评分搜索(Scoring and Searching, SS)的学习算法和混合学习算法<sup>[5]</sup>。CB 算法认为结构学习问题是约束满足问题,通过检验条件独立性(Conditional Independence, CI)来构建结构。SS 算法则把结构学习问题表述为结构优化问题,利用评分函数评价每个候选结构,然后搜索出分数最高的结构。CB 算法的实现比较简单,但是高阶 CI

到稿日期:2015-01-14 返修日期:2015-04-10 本文受国家自然科学基金资助项目(61305058, 61300139),厦门科技计划基金资助项目(3505Z20133027),华侨大学科研基金资助项目(11Y0274, 12HJY18),中央高校基本科研基金资助项目(11J0263)资助。

SEIN Minn(1990-),男,硕士生,主要研究方向为数据挖掘及其应用, E-mail: 1525441329@qq.com; 傅顺开(1978-),男,博士,讲师,主要研究方向为数据挖掘和机器学习, E-mail: fusk@hqu.edu.cn.

检验需要充足的数据才能保证结果的可靠性。SS算法虽然是目前使用最为广泛的结构学习算法,但其收敛速度慢,计算复杂。混合学习算法把CB算法和SS算法结合起来,充分利用两者的优点,可显著提高学习速度并能处理大型网络。混合学习算法一般先使用CI检验构造变量序列或缩减搜索空间,然后再通过SS算法进行结构学习。

除了这3类主要的结构学习算法外,还有动态规划结构学习、基于遗传算法的结构学习、模型平均结构学习和不完备数据集的结构学习等算法<sup>[6]</sup>。

本文提出一种新的CB类算法,它基于一系列的局部邻域结构推导来完成全局BN结构的学习。新算法需要的假设与同类的经典算法PC(Peter & Clark)一致,且继承了PC算法的最大优点即推导过程中优先执行低阶CI测试,这可保证算法的实用价值。与PC算法“平等”对待所有候选CI测试不同,新算法根据一定策略对候选CI测试进行排序,优先执行更可能带来“效果”的CI测试,避免了无效计算。因此新算法被命名为APC(Accelerated PC)。基于多个网络(问题)的实验研究验证了新算法与PC算法在推导效果上一致,但具有显著的时间效率优势。例如,在面对一个200个节点的网络时,APC较PC算法节省了约50%的计算量。

## 2 相关研究

### 2.1 符号表示

采用粗体英文字母来表示变量集合(比如 $\mathbf{X}$ ),非粗体的字母表示单个的变量(比如 $X$ ),小写英文字母表示变量值(比如 $X=x$ )。对于一个分布 $\rho$ ,采用 $X \perp_{\rho} Y | Z$ 来表示 $X$ 与 $Y$ 关于条件集 $Z$ 相互独立。若条件集 $Z = \emptyset$ (为空),用 $X \perp_{\rho} Y$ 来表示。为了简化表示,当考虑单个变量时,忽略标准的集合表示。比如,采用 $X \perp_{\rho} Y | Z$ 表示 $\{X\} \perp_{\rho} \{Y\} | \{Z\}$ 。类似地,使用 $X \perp_G Y | Z$ 来表示在概率图模型 $G$ 中的条件独立关系。

### 2.2 贝叶斯网络和相关基础

给定问题域 $U = \{X_1, \dots, X_n\}$ 上的联合分布 $\rho$ ,对应的BN是一个二元组, $B = \langle G, \Theta \rangle$ ;  $G = \langle V, A \rangle$ 是一个有向无环图, $V (=U)$ 和 $A$ 分别表示节点和边集合,包含了BN的定性和直观描述。对于任意的节点 $X \in V$ ,与其关系紧密的父、子和配偶节点分别记为 $Pa(X)$ 、 $Ch(X)$ 和 $Sp(X)$ 。例如图1中的 $Pa(X_8) = \{X_3, X_5\}$ ,  $Ch(X_8) = \{X_4, X_7\}$ ,  $Sp(X_8) = \{X_2, X_5\}$ 。 $\Theta$ 为参数集合,包含了诸如 $\Theta_{x|Pa(x)} = p(x|Pa(x))$ 这样的条件概率分布,它定量地给出当父节点集合取值为 $Pa(x)$ 时它们的共同子节点 $X$ 取 $x$ 的条件概率。若 $Pa(X) = \emptyset$ ,参数为先验概率 $p(x)$ 。

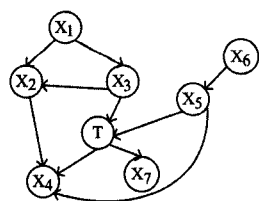


图1 贝叶斯网络例子

一个BN所表示的(全局)联合分布可根据局部结构因式分解为:

$$p(X_1, \dots, X_n) = \prod_{i=1}^n p(X_i | Pa(X_i)) \quad (1)$$

定义1( $d$ -分割, dependence-separate, or  $d$ -separate) 有

向图 $G$ 中的条件独立关系 $X \perp_G Y | Z$ 表示节点 $X$ 和 $Y$ 被 $Z$  $d$ -分割,相应的 $Z$ 称作关于 $X$ 和 $Y$ 的 $d$ -分割集。

定义2(忠实, Faithfulness) 一个贝叶斯网络 $B$ 和一个联合分布 $\rho$ 是相互忠实的,当且仅当任意的 $X \perp_G Y | Z$ 有相应的 $X \perp_{\rho} Y | Z$ <sup>[7]</sup>。

当忠实性满足时, $d$ -分割概念桥接了概率视图和图视图下的条件独立关系:(1) $X \perp_{\rho} Y | Z$ 同 $X \perp_G Y | Z$ 是一一对应的;(2) $d$ -分割集“割断”了 $X$ 和 $Y$ 之间的所有有向路径。 $d$ -分割是CB类学习算法的核心基础——只要能找到任意节点对 $X$ 和 $Y$ 的 $d$ -分割集,即可判断不相邻。

定理1 给定忠实假设, $X$ 和 $Y$ 相邻当且仅当不存在 $Z$ 使得 $X \perp_{\rho} Y | Z$ 且 $X, Y \notin Z$ <sup>[7]</sup>。

从定理1容易推出引理1。

引理1 给定忠实假设,如果存在一个不包含 $X$ 和 $Y$ 的集合 $Z$ 使得 $X \perp_{\rho} Y | Z$ ,那么 $X$ 和 $Y$ 不相邻(即边 $X-Y$ 不存在)。

定理2 给定忠实假设, $X \perp_{\rho} NonDes(X) | Pa(X)$ ,即 $X$ 和所有 $X$ 的非后代 $NonDes(X)$ 关于 $Pa(X)$ 条件独立<sup>[7]</sup>。

换句话说, $X$ 和 $NonDes(X)$ 被 $Pa(X)$  $d$ -分割。

忠实假设是本文提出的新算法所依赖的假设,而 $d$ -分割和 $d$ -分割集概念是算法设计的重要依据。

定义3(骨架, Skeleton) 一个贝叶斯网络 $X$ 的所有边去掉方向后的无向图即为该网络对应的骨架。

不同的贝叶斯网络可能含有相同的骨架。几乎所有的CB类算法都是先输出骨架信息,而后再依据一定的规则确定方向。本研究提出的算法也不例外。

### 2.3 CB类算法研究进展

理想的学习目标是从样本数据还原一个和真实分布 $\rho$ 完全一致的模型 $M$ 。但这个目标限于两个原因而一般无法实现:计算复杂度和有限样本,而后者影响更大,因为样本不足时只能反映不完整的分布关系,且这是算法无法改变和逾越的。

CB类算法可分为两大类。绝大多数算法并不要求指定节点(变量)的次序,且基于两个基本假设:忠实和正确的CI测试。此类算法较早期的工作有SGS算法<sup>[7]</sup>,但最经典的当属SGS的改进版本即PC算法<sup>[7]</sup>。PC算法的设计策略使得该算法能尽可能避免执行高阶CI测试。IC算法<sup>[8]</sup>是另一个类似PC的工作,由Pearl等同期提出。GSBN算法(原始和随机版本)<sup>[9]</sup>的设计参考了SGS算法,但其作者将全局结构推导分解为各个节点邻域的推导。具体来说,通过分别学习各个节点的马尔科夫毯(这个过程在原文中称为GSMB),之后“拼接”局部结构获得全局贝叶斯网络。GSBN算法虽然时间复杂度低,但需要十分庞大的数据集才能实现可接受的学习效果,因此不利于应用。

有一部分CB算法需要指定节点的顺序,这要求更严格的假设,或辅以更多专家的人工判断。节点顺序信息被利用后可较大幅度降低搜索空间复杂度,从而提高学习的时间效率。早期的工作有Wermuth-Lauritzen<sup>[10]</sup>,而最著名的当属TPDA算法<sup>[11]</sup>,但是TPDA需要一个新的假设,即所谓的单调忠实(Monotone Faithfulness)。基于该假设,TPDA算法可以实现 $O(n^4)$ 的复杂度水平。PMMA算法<sup>[12]</sup>同样需要单调忠实假设,并取得了类似的时间复杂度。然而,Chickering等

证明了单调忠实是一个糟糕的假设<sup>[13]</sup>。因此,此类算法并不在本研究的讨论和比较范围。

以上提及的CB算法一般输出为目标网络的骨架,需要额外的规则来完成对边方向的确定。一般来说,这又可以分为两个子步骤:(1)检测和确定 $v$ -结构(即 $X \rightarrow Y \leftarrow Z$ );(2)剩余边的方向的确定,原则是不会形成有向环。

### 3 APC 算法描述

总体来看,APC算法采用“分而治之”策略,依赖一系列的局部邻域的推导从而完成全局BN的学习。在推导邻域时,APC算法参考了PC算法的后向选择(Backward Selection)策略,即先假设当前节点 $X$ 和其他节点都相连,继而根据引理1依次寻找 $X$ 和任意邻居 $Y$ 是否存在 $d$ -分割集;一旦发现,立即删除相关边。相比直接基于定理1,这样做往往可以节省可观的计算量。而在寻找是否存在满足条件的 $d$ -分割集时,APC算法基于更有效的策略(见3.1节讨论),从而进一步节省了计算量。

#### 3.1 启发式搜索策略

CB类算法高度依赖CI测试,而实际应用中高阶CI测试的可靠性受限于训练样本规模,这将直接影响学习效果。因此,设计CB类算法时需要尽量避免高阶CI测试,以保证此类算法的实用价值。对于两个不同的CB类算法A和B,当给定相同的样本集时,若算法A输出的网络结构的精确度高于算法B,则称算法A的数据效率(Data Efficiency)更高,即实用性较佳。

现有的CB类算法中,数据效率最高的当属PC算法,因为其推导过程优先执行低阶CI测试。PC算法先假设 $\forall X, Y \in U$ 相连,随后检查每对节点 $X$ 和 $Y$ 是否存在 $d$ -分割集,若存在,则删除边 $X-Y$ (依据引理1),否则保留之。PC算法设计的巧妙之处是从空集开始检查各个候选集合是否为 $d$ -分割集,该策略允许找到尽可能小的 $d$ -分割集,并通过尽可能低阶的CI测试来决定是否删除边 $X-Y$ 。

为了寻找 $d$ -分割集,PC算法会构建所有可能的候选集合并将其逐一取出检验,直到发现符合条件的集合(即结束搜寻),或完成所有候选集合的检验。如图2所示,若要寻找 $X$ 和 $Y$ 之间是否存在包含一个变量的 $d$ -分割集,候选集合有 $\{Z\}, \{M\}, \{O\}, \{P\}, \{N\}$ 和 $\{Q\}$ ,由于PC算法随机挑选, $\{Z\}$ 可能排在末尾被选中,这样所执行的6个CI测试就有5个是无效的。如果 $\{Z\}$ 能第一个被挑选,则仅需执行一个CI测试。图2的BN中, $\{Z\}$ 不但是 $X$ 和 $Y$ 的 $d$ -分割集,还是 $X$ 和 $M, X$ 和 $N, X$ 和 $O$ 等的 $d$ -分割集。 $\{Z\}$ 之所以能“割断”这么多对节点,是因为 $Z$ 靠近 $X$ ,所有从 $X$ 出发到其他节点的有向路径都必须经过 $Z$ 。对于与 $Z$ 相比离 $X$ 稍远些的 $M, Y$ 和 $N$ ,它们则只能分别“割断” $X$ 与 $O, P$ 和 $Q$ 的有向路径;而距离 $X$ 最远的 $O, P$ 和 $Q$ ,则没有机会“割断”任何路径。本项目正是基于该拓扑信息,着重研究如何找到距离 $X$ 近的节点,并且认为基于距离 $X$ 近的节点所组成的集合更可能成为 $d$ -分割集,即应该优先基于这些集合执行CI测试。

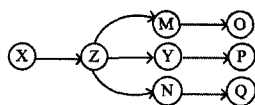


图2 寻找 $d$ -分割集

#### 3.2 算法流程描述

APC算法的过程如表1所列,它应用到3个关键数据结构: $A_{Ind}$ 、 $A_{Del}$ 和 $A_{Can}$ ;学习过程依赖于过程即LPC算法(见表2)完成关于每个 $X \in U$ 的邻域的推导:

1.  $A_{Ind}$ 用于存储确定的边, $A_{Del}$ 用于存储确定应不存在的边,都被初始化为空集(第1行);
2.  $A_{Ind}$ 在每个新的迭代中都被初始化为包含当前的 $X$ 和其他所有变量 $U \setminus \{X\}$ 的边,但要从中删除 $A_{Del}$ 中的边(第3行);
3. LPC算法中将判断且删除 $A_{Ind}$ 中的假正(False Positive,即实际不应该存在的)边,被删除的边将被添加入 $A_{Del}$ (第4行);
4. 算法第2-5行的循环结束后, $A_{Del}$ 中包含目标贝叶斯网络的完整骨架,随后应用一系列规则确定边方向(第6行)。

表1 APC算法伪代码

```

APC:BN 结构推导
输入:数据集 D
输出:BN 结构
1.  $A_{Ind} = A_{Del} = \emptyset$ ;
2. for ( $\forall X \in U$ ) do
3.    $A_{Ind} = A_{Ind} \cup \{X-Y \mid \forall Y \in U \setminus \{X\}\} \setminus A_{Del}$ ;
4.   LPC( $X, D, A_{Ind}, A_{Del}$ );
5. end for
6. Apply orientation rules on  $A_{Ind}$ ;
7. return  $A_{Ind}$ ;
  
```

从全局来看,APC算法执行的是前向选择(Forward Selection)的搜索,即随着学习的进行不断获得属于目标BN的边信息。

LPC算法是APC算法的核心组成,担负“去伪存真”的任务,它所执行的是后向选择(Backward Selection)的搜索。根据引理1,如果找到 $T$ 和候选邻居 $X$ 的 $d$ -分割集(第7和19行),则分别执行第8-10行以及第19-23行部分,它们包括3个共同的操作:(1)将 $X$ 从候选邻居集 $V_{CanPC}$ 中删除;(2)从 $A_{Ind}$ 中删除边 $X-T$ ;(3)往 $A_{Del}$ 中添加边 $X-T$ 。由于第19-23行对应 $d$ -分割集 $S$ 不为空的场景,因此增加一个操作:对新发现的分割集 $S$ 中的每个变量 $Y$ ,更新其相关的频率信息 $Freq_Y$ (第23行)。第23行的设计是APC算法的核心之处——为判断 $X$ 和 $T$ 的距离远近“收集”依据:新发现的 $d$ -分割里的每个变量 $X$ 的 $Freq_X$ 集分别增加1。经过一系列CI测试,如果 $Freq_X > Freq_Y$ ,则认为 $X$ 比 $Y$ 更接近 $T$ 。

函数GenerateSS负责基于 $Freq_Y$ 信息生成供CI测试的候选条件集合SS。首先,它基于 $V_{CanPC}$ 生成一系列的候选 $S$ ,且 $|S| = css$ 。其次,对所有的 $S$ 从大到小排序,排序参考数值 $\sum_{Y \in S} Freq_Y$ 。这样,排在靠前的 $S$ 比排在靠后的包含更多靠近 $T$ 的节点;根据3.1节的讨论,这样的集合的确更可能“截断” $T$ 到其他节点的路径。

生成的候选 $d$ -分割集的集合SS在第17行处被逐一取出并应用到CI测试中。由于在GenerateSS中对候选 $d$ -分割集进行了排序,有可能很快找到满足条件的 $d$ -分割集;而一旦发现 $T$ 和 $X$ 关于 $S$ 条件独立,则忽略剩余的分割集候选集合(第23行),从而减少无效CI测试的执行。第5节的实验研究证实了这一点,而第4节将证明这样的设计仍能保证算法的正确性。

表2 LPC算法的伪代码

```

输出:当前节点  $T$ 、数据集  $D$ 、候选邻居集  $A_{Ind}$ 、已知不存在的边集  $A_{Del}$ 
输出:更新的  $A_{Ind}$ 
1.  $V_{CanPC} = \{X | (T-X) \in A_{Can}\}$ ;
2.  $css = 0$ ;
3.  $Freq_X = 0, \forall X \in U$ ;
4. while ( $|V_{CanPC}| > css$ ) do
5.   if ( $css < 1$ ) then
6.     for ( $\forall X \in V_{CanPC}$ ) do
7.       if ( $I_D(T, X) \geq 1 - \epsilon$ ) then
8.          $V_{CanPC} = V_{CanPC} \setminus \{X\}$ ;
9.          $A_{Ind} = A_{Ind} \setminus \{T-X\}$ ;
10.         $A_{Del} = A_{Del} \cup \{T-X\}$ ;
11.       end if
12.     end for
13.     Sort  $X \in V_{CanPC}$  in descending order according to their dependency on  $T$ ;
14.   else
15.     for ( $\forall X \in V_{CanPC}$ ) do
16.        $SS = \text{GenerateSS}(V_{CanPC}, css, Freq)$ ;
17.       for ( $\forall S \subseteq SS$ ) do
18.         if ( $I_D(T, X) \geq 1 - \epsilon$ ) then
19.            $V_{CanPC} = V_{CanPC} \setminus \{X\}$ ;
20.            $A_{Ind} = A_{Ind} \setminus \{T-X\}$ ;
21.            $A_{Del} = A_{Del} \cup \{T-X\}$ ;
22.            $\forall Y \in S, Freq_Y ++$ ;
23.         break;
24.       end if
25.     end for
26.   end for
27. end if
28.  $css ++$ ;
29. end while

```

### 3.3 算法特点小结

同PC算法,APC也属于CB类算法,且不需要节点排序信息。APC算法的学习由一系列的局部推导协助完成,综合了前向和后向选择的优点。局部学习的后向选择策略有助于找到尽可能小的 $d$ -分割集,能最大程度规避CB类算法面临的高阶CI测试的可靠性问题。而对候选 $d$ -分割集的排序处理则有利于优先执行有效的CI测试,提高学习效率。因此,APC算法兼顾了CB类算法的两大方面——可靠性和效率。

APC算法虽然也基于一系列的局部推导,但与GSBN算法相比有显著的不同。首先,GSBN的局部搜索目标是各节点的马尔科夫毯,而APC推导的目标仅仅是马尔科夫毯的一部分,即(直接相连的)父、子节点。其次,GSBN所执行的CI测试可能将整个马尔科夫毯甚至更大的节点集合作为条件集合;APC算法中所执行的CI测试的条件集合相比要小许多,因此测试结果可靠性更高。考虑到CB类算法的级联错误效应,GSBN的实际应用效果并不佳。

## 4 算法证明和分析

APC算法基于忠实假设和正确的CI测试而设计,所以下面证明都默认这两个假设满足。

### 4.1 算法证明

CB类算法一般都是先推导网络骨架,再应用一定的规则完成边方向的确定,而第一个任务相对要困难得多。因此,我们重点证明APC算法能够正确完成骨架的推导。

**引理2** LPC算法不会删除本该存在的边 $T-X$ 。

**证明:**假设针对 $T$ 调用LPC,初始化后的 $A_{Ind}$ 包含边 $T-X$ ,且这条边属于目标贝叶斯网络。如果边 $T-X$ 在LPC中

被删除,意味着存在某个CI测试没有通过(第7或18行),即存在使 $T$ 和 $X$ 条件独立的集合。这与定理1相悖,故真实的边 $T-X$ 不会被LPC误删。

**引理3** LPC( $T$ )中确定删除的边 $T-X$ 可以被放入 $A_{Del}$ 。

**证明:**根据独立性交换律,如果 $T \perp X$ ,那么 $T \perp X$ 同样成立。

一旦某条边被检测为假正而放入 $A_{Del}$ ,这样的边在未来将无需重复检测(见算法1的第3行)。这样的设计可以有效避免无用计算,提高学习效率。

**引理4** 调用LPC( $T$ )后, $A_{Ind}$ 中可能包含目标贝叶斯网络不存在的边 $T-X$ 。

**证明:**(1)引理1告诉我们所有和 $T$ 直接相连的节点将一直保留在 $V_{CanPC}$ 中,包括 $Pa(X)$ 。(2)定理2告诉我们由于 $Pa(X)$ 的存在,所有的非后代节点 $NonDes(X)$ 都会被从 $V_{CanPC}$ 中删除(当 $css = |Pa(X)|$ 时),相关的边也会被从 $A_{Ind}$ 中删除。(3)只有 $T$ 和部分后代节点之间的边在某些场景下可能无法被正确识别和删除。

虽然部分假正边 $T-X$ 可能在LPC( $T$ )中无法被正确删除,但引理5将告诉我们由于还将调用LPC( $X$ ),这样的假正边在LPC( $X$ )中能够被正确删除。

**引理5** LPC( $X$ )中没有被删除的不存在边 $X-Y$ 将在LPC( $Y$ )中被删除。

**证明:**引理4告诉我们只有 $X$ 和部分后代 $Y$ 之间的边可能无法被正确识别,所以只需证明这部分边在LPC( $Y$ )中能够被识别而删除。因为 $Y$ 是 $X$ 的后代,可知 $X$ 是 $Y$ 的非后代,根据引理4的证明可知 $Y-X$ 将从 $A_{Ind}$ 中删除。

因此,目标贝叶斯网络中不存在的边 $X-Y$ 即使由于算法设计的缘故而被引入 $A_{Ind}$ 中,在分别调用LPC( $X$ )和LPC( $Y$ )后,其也将最终被删除。

**引理6** LPC( $X$ )执行完毕后保留在 $A_{Ind}$ 中的真实边 $X-Y$ 在调用LPC( $Y$ )后将仍然保留。

**证明:**如果 $X \not\perp Y$ ,那么易得 $Y \not\perp X$ 。根据引理2,这样的边在调用LPC( $Y$ )后仍然将保留在 $A_{Ind}$ 中。

**定理3** APC算法能够正确推导出贝叶斯网络的骨架。

**证明:**(1)按APC的设计,任意节点对 $X$ 和 $Y$ 都会被假设相邻,因此不会错过任何一条可能存在的边;(2)引理2和引理6保证了正确的边不会被误删除;(3)引理3和引理5保证了(1)中引入的假正边都将被识别并删除。因此,在APC执行完毕后, $A_{Ind}$ 中将包含且仅包含目标贝叶斯网络的骨架。

### 4.2 算法复杂性分析

算法复杂性将直接影响其实用性。CB类算法的复杂性包括两个方面:时间(需要执行多少个CI测试)和测试维度(所执行的CI测试的维度)。

若不考虑对候选 $d$ -分割集优先级别的判断,APC算法将等同于PC算法。而APC算法的实际时间效率受所面对的网络拓扑影响较大,最理想情况是LPC过程中每一条假正边的判断仅需执行一次CI测试。随着网络结构复杂度的提高,APC算法所需的计算量也将增加,因为任何两个节点 $X$ 和 $Y$ 之间的有向路径将增多,而“割断”它们将不得不依赖更大的 $d$ -分割集。

高阶CI测试对CB类算法是致命的,不但影响当前测试结果的可靠度,而且将带来级联错误。幸运的是,APC算法

能够尽可能规避高阶测试。

**定理 4** APC 算法能够找到任意对节点  $X$  和  $Y$  的最小  $d$ -分割集。

证明:(反证法)假设 LPC 算法中找到的  $X$  和  $Y$  的  $d$ -分割集为  $S$ ,而实际最小的  $d$ -分割集为  $S^*$ ,且  $S^* \subset S$ 。这意味着当  $css = |S^*|$  时,第 7 或 18 行的 CI 测试失败,这与假设不符。

定理 4 显示 APC 能尽可能规避高阶 CI 测试,这对于提高学习效果和降低学习成本都有帮助。

## 5 实验研究

### 5.1 实验设计

考虑到关注重点是结构学习的质量,选取已知真实结构的问题,包括两个经典网络(关于“非典”建模的 Asia 网络与用于病人监控和诊断的 Alarm 网络)和两个基于 Weka 的 BN 软件包随机生成的网络(Test100 和 Test200)(见表 3)。

表 3 实验数据集

数据集	节点数	边数值	变量取值
Asia	8	8	2
Alarm	37	46	2~4
Test100	100	120	2
Test200	200	250	2

算法比较上只选择了同类最经典的 PC 算法,考虑到:(1)PC 算法被引用率最高,且在诸多论著的实验研究中显示了明显的优势;(2)APC 算法和 PC 算法所需的假设一致,具有可比性;(3)GSMB 算法依赖 GSMB,而 Fu 等关于马尔科夫毯推导算法的比较研究已经指出,比 GSMB 更优的 IAMB 算法的学习效果远差于 PC 算法<sup>[14,15]</sup>,故可间接推导 GSMB 算法实用性不佳,没有必要对比;(4)TPDA 和 PMMA 所要求的单调忠实假设被证明不切实际,故不在考虑范畴。

APC 算法的实现基于 Weka 框架(v3.7),PC 算法则基于 Weka 的 BN 软件包的实现版本。算法所基于的 CI 测试也是基于 Weka 的实现的。

从以下 4 个维度分别对算法性能进行评估:

1. 面向知识发现的学习效果。知识发现任务关注每条边,无论是强关系还是弱关系边,而 Hamming 距离适合衡量这样的学习效果。Hamming 距离被定义为将学习所得网络转变为真实网络所需的操作数,而允许的操作包括添加边、删除边和翻转边的方向。

2. 面向密度估计的学习效果。与知识发现任务相比,密度估计任务基于全局网络进行评价,故在个别弱关联边上的错误并不会太影响评价得分。实验中选择基于经典的 BDeu 评分方法。

3. 学习效率。学习效率是基于 CI 测试的加权统计<sup>[16]</sup>。针对一个 CI 测试  $I_D(X, Y|Z)$ ,它的权重为  $2 + |Z|$ ,反映了统计测试复杂度与涉及的变量(数)成正比。而一个算法的复杂度则为整个学习过程所执行的所有这样的 CI 测试的权重之和。这是基于 CI 测试的统计学习算法的标准比较方法,其最大优点是机器无关性。

4. 网络密度对算法的影响。对于相同节点数量的网络,边数越多说明网络密度越高。在 Test100 网络基础上,通过不断随机地增加边,观察 APC 和 PC 两个算法的相对性能,同样基于以上 3 个维度。

### 5.2 面向知识发现的学习效果评估

图 3 是关于 APC 和 PC 的学习效果。(1)4 个实验中,随着学习样本的增多,两个算法的学习精准度都得到了提高。(2)随着问题规模增大,虽然错判的边数增加,但相对错判率在下。例如在 Test200 实验中,当有 5000 个学习样本时,只有约 4 条边被误判,即约 1.6%(4/250)的错误率。因此,两个 CB 类算法的学习效果都比较理想。(3)APC 算法的学习效果和 PC 算法相当,间接验证了新算法的正确性。

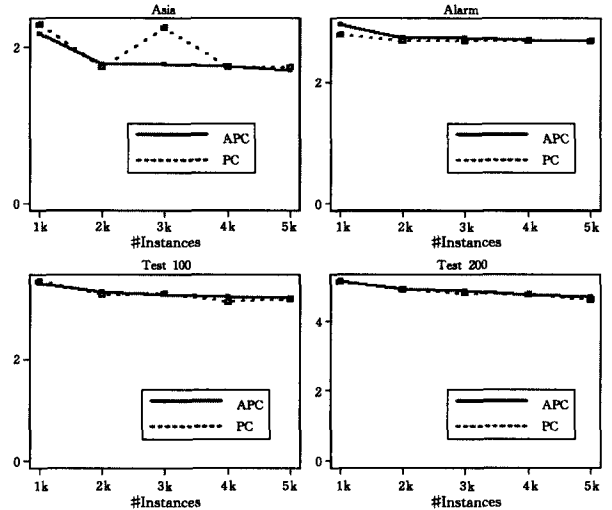


图 3 基于 Hamming 距离的学习质量度量

### 5.3 面向密度估计的学习效果评估

图 4 是从面向密度估计学习任务方面对 APC 和 PC 算法进行学习效果的对比。(1)同样在 4 个实验问题中,随着学习样本的增多,两个算法的学习效果都得到了提高,且提高的速度显著;(2)APC 算法的学习效果和 PC 算法相当,进一步验证了新算法的正确性;(3)相对性能趋势与面向知识发现的学习效果一致。

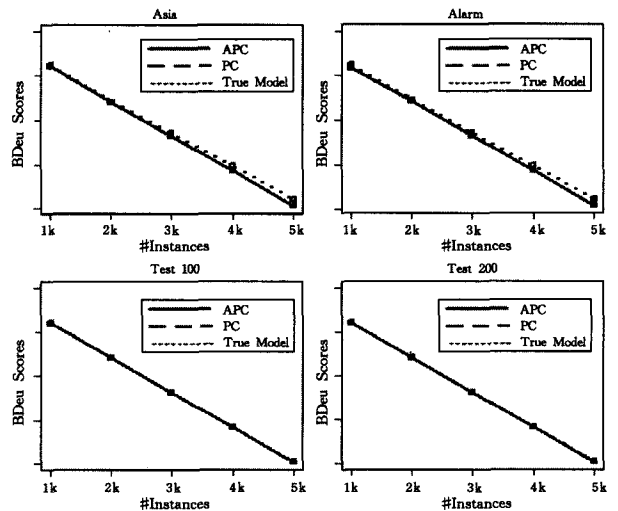


图 4 基于 BDeu 的学习质量度量

### 5.4 学习效率评估

图 5 是关于 APC 和 PC 实际学习效率的对比。(1)4 个实验问题中,随着学习样本的增多,两个算法都需要消耗更多的计算资源。这是统计学习的典型特征:更多的样本允许更多的(较高维度的)CI 测试变得可靠,这使得搜索可以走得更远(意味着更准确的输出,如图 3 和图 4 所示)。(2)虽然两者

所需的计算量都增加,但 APC 增加的速度要慢于 PC 算法。  
 (3) APC 较 PC 可以节省可观的计算资源,例如表 4 中 APC 较 PC 算法能节省约 52% 的计算量。

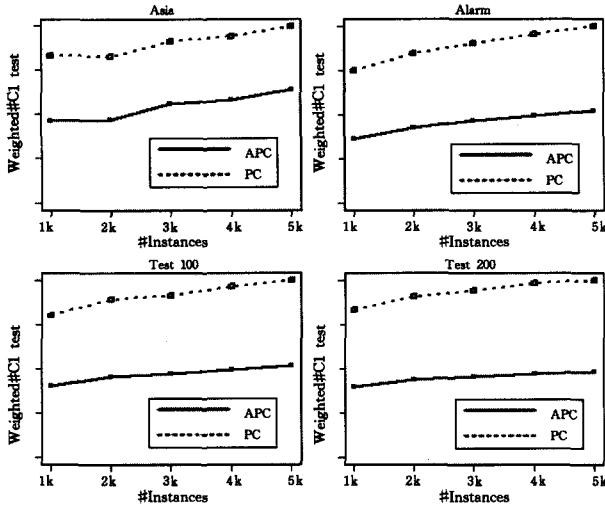


图 5 基于加权 CI 测试数的学习效率度量

表 4 Test100 网络实验中的加权 CI 测试数比较

样本集	PC	APC	节省
1k	137526.8	65404.8	52.4%
2k	149362.0	72051.8	51.7%
3k	155262.7	74495.2	52.0%
4k	161515.1	77016.6	52.3%
5k	164929.4	79445.9	51.8%

### 5.5 网络密度对计算效率的影响

保持 Test100 网络 100 个节点规模,通过仿真程序随机生成 130、150、170、190 和 210 条边的网络,继而基于生成的网络再随机生成 1k、2k、3k 和 4k 大小的样本集供实验。实验显示,APC 和 PC 算法无论是面对知识发现还是密度估计任务,所达到的学习质量与图 3 和图 4 类似,为节省篇幅此处就不再展示。图 6 为两个算法在面对不同密度网络时的学习效率比较:(1)随着密度增大(网络变复杂),两个算法都需要更多的计算量;(2)两个算法计算量的增速相当,因此 APC 始终保持对 PC 算法一致的效率优势。

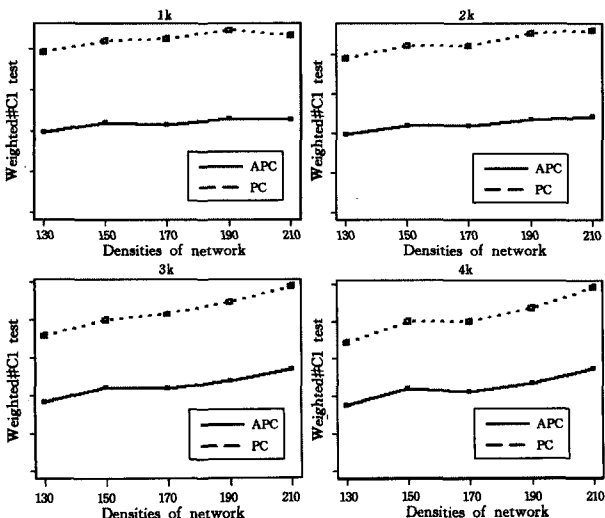


图 6 针对 Test100 网络实验提高网络密度对计算效率的影响 (仍然基于加权 CI 测试数)

想工具,而结构学习是应用贝叶斯网络的基础任务。提出一种新的基于约束学习的算法,其不但继承了 PC 算法的高数据效率特性,而且通过动态调整 CI 测试顺序以优先执行能带来效果的 CI 测试,间接节省了计算量。实验显示,APC 较 PC 算法在计算效率方面能带来可观的提升。未来研究可集中在进一步挖掘拓扑信息以加速学习进程,例如当前研究中学习 X 邻域时所获得的 d-分割集信息在本 LPC 过程结束时将被抛弃,而这个信息是针对一对节点 X 和 Y 而言的。若能将其“带入”关于 Y 的 LPC 中,可避免从头开始搜集相关的拓扑信息,有望进一步节省计算量。

### 参考文献

- [1] Liang Jie, Cai Qi, Chu Zhu-li, et al. Reactor water system diagnostic Bayesian network construction Legislation and application [J]. Atomic Energy Science and Technology, 2013(10): 1840-1844(in Chinese)  
梁洁,蔡琦,初珠立,等.反应堆补水系统诊断贝叶斯网络的建立和应用[J].原子能科学技术,2013(10):1840-1844
- [2] Ren Jia, Du Wen-cai, Bai Yong. Adaptive Bayesian network based inference UAV mission decisions [J]. Systems Engineering Theory and Practice, 2013, 33(10): 2575-2582(in Chinese)  
任佳,杜文才,白勇.基于贝叶斯网络自适应推理的无人机任务决策[J].系统工程理论与实践,2013, 33(10): 2575-2582
- [3] Barbaros Y, Zane B P, Todd E R, et al. Combining data and meta-analysis to build Bayesian networks for clinical decision support[J]. Journal of Biomedical Informatics, 2014, 52: 373-385
- [4] Liu Jian-wei, Li Hai-en, Luo Xiong-lin. Research progress of probabilistic graphical model learning technology [J]. Journal of Automatic, 2014, 40(6): 1025-1044(in Chinese)  
刘建伟,黎海恩,罗雄麟.概率图模型学习技术研究进展[J].自动化学报,2014, 40(6): 1025-1044
- [5] Daphne K, Nir F. Probabilistic graphical models: principles and techniques[M]. MIT Press, 2009
- [6] Guo Tong, Lin Feng. Hybrid genetic fish group based Bayesian network structure learning algorithm [J]. Journal of Zhejiang University(Engineering Science), 2014, 48(1): 130-135(in Chinese)  
郭童,林峰.基于混合遗传鱼群算法的贝叶斯网络结构学习[J].浙江大学学报(工学版),2014, 48(1): 130-135
- [7] Peter S, Clark G, Richard S. Causation, Prediction, and Search [M]. A Bradford Book, 2001
- [8] Judea P, Verma T S. A Theory of Inferred Causation[C]//Proceedings of 2nd International Conference on Principles of Knowledge Representation and Reasoning, 1991
- [9] Dimitris M, Thrun S. Bayesian Network Induction via Local Neighborhoods[C]//Proceedings of the 12th Neural Information Processing Systems (NIPS). MIT Press, 1999
- [10] Wermuth N, Lauritzen S. Graphical and recursive models for contingency tables [J]. Biometrika, 1983, 72: 537-552
- [11] Jie C, David A B, Weiru L. Learning Belief Networks from Data: An Information Theory Based Approach[C]//Proceedings of the 6th ACM International Conference on Information and Knowledge Management (CIKM). 1997: 325-331

结束语 贝叶斯网络是一种对不确定问题进行建模的理

(下转第 272 页)

表3 优势类的分布情况及区域归属

比较: $ [x_i]_{\beta}^{\geq} $	比较: $ [x_i]_{\beta}^{\leq} $	比较: $ [x_i]_{\beta}^{\geq} \cap X $	$x_i$ 优势 类分布	归属 区域	比较 次数	变量 个数
$< \frac{k}{\beta}$	$> 2k$	$\geq  [x_i]_{\beta}^{\geq}  - k$	Null	Pos	4	1
$< \frac{k}{\beta}$	$\leq 2k$	$> k$	20	Pos	3	0
$< \frac{k}{\beta}$	$> 2k$	$\leq k$	18	Neg	3	0
$< \frac{k}{\beta}$	$\leq 2k$	$<  [x_i]_{\beta}^{\geq}  - k$	2	Neg	4	1
$< \frac{k}{\beta}$	$> 2k$	$(k,  [x_i]_{\beta}^{\geq}  - k)$	Null	Ubn	4	1
$< \frac{k}{\beta}$	$\leq 2k$	$[[x_i]_{\beta}^{\geq}  - k, k]$	1,6,7, 9,11	Lbn	4	1
$\geq \frac{k}{\beta}$	Null	$\geq (1-\beta)  [x_i]_{\beta}^{\geq} $	Null	Pos	3	2
$\geq \frac{k}{\beta}$	Null	$\leq \beta  [x_i]_{\beta}^{\geq} $	3,4,5,8, 10,12,13, 14,15,16, 17	Neg	2	1
$\geq \frac{k}{\beta}$	Null	$(\beta  [x_i]_{\beta}^{\geq} , (1-\beta)  [x_i]_{\beta}^{\geq} )$	19	Ubn	3	2

两种方法所求结果是一致的,但是其求解过程中的时间和空间复杂性却不一样,利用定理 1 求解的时间复杂性  $T_1 = 80$ ,空间复杂性  $S_1 = 40$ ;而利用定理 4 求解的时间复杂性  $T_2 = 54$ ,空间复杂性  $S_2 = 19$ 。该例说明当有大量的数据需要处理时,选择定理 4 这一方案求解,在保持实验结果不变的情况下,在计算时间和存储空间上具有明显的优势。

**结束语** 本文在序信息系统中通过“逻辑与”和“逻辑或”将变精度粗糙集与程度粗糙集融合起来,建立了一种新的粗糙集模型,而后深入研究了模型的数学性质。最后通过两种方法对学生成绩这一现实例子进行了分析,通过对方法的比较可以知道,正是通过本文的工作对粗糙集区域进行了更精确的刻画,使得在实际工程应用中能从时间和空间两方面来简化求解。

参 考 文 献

[1] Dembczyński K, Pindur R, Susmaga R. Dominance-based rough set classifier without induction of decision rules[J]. Electronic Notes Theory Computer Science, 2003, 82: 84-95

[2] Pawlak Z. Rough Sets [J]. International Journal of Computer and Information Sciences, 1982, 11(5): 341-356

[3] Shen Jin-biao, Lv Yue-jin. Promotion for rough set of variable precision and extent of rough set[J]. Computer Engineering and Applications, 2008, 44(36): 45-47 (in Chinese)

申锦标,吕跃进. 变精度与程度粗糙集的一种推广[J]. 计算机工程与应用, 2008, 44(36): 45-47

[4] Xu Wei-hua, Liu Shi-hu, Wang Qiao-rong. The First Type of Grade Rough Set Based on Rough Membership Function[C]// 2010 Seventh International Conference System and Knowledge Discovery(FSKD2010). 1922-1926

[5] Xu Wei-hua. Ordered Information Systems and Rough Set[M]. Beijing: Science Press, 2013 (in Chinese)

徐伟华. 序信息系统与粗糙集[M]. 北京: 科学出版社, 2013

[6] Yu Jian-hang, Xu Wei-hua. The rough set based on logical conjunction operation of variable precision and grade in ordered information system[J]. Fuzzy Systems and Mathematics, 2015, 29(4): 146-152 (in Chinese)

余建航, 徐伟华. 序信息系统下基于精度与程度的“逻辑与”的粗糙集[J]. 模糊系统与数学, 2015, 29(4): 146-152

[7] Yu Jian-hang, Xu Wei-hua. The rough set based on logical disjunct operation of precision and grade in ordered information system[J]. Journal of Frontiers of Computer Science and Technology, 2015, 9(1): 112-118 (in Chinese)

余建航, 徐伟华. 序信息系统下变精度与程度的“逻辑或”粗糙集[J]. 计算机科学与探索, 2015, 9(1): 112-118

[8] Yao Y Y, Lin T Y. Generalization of rough sets using modal logics[J]. Intelligent Automatic and Soft Computing, 1996, 2: 103-120

[9] Ziarko W. Variable precision rough set model [J]. Journal of Computer and System Sciences, 1993, 46(1): 39-59

[10] Zhang Wen-xiu, Wu Wei-zhi, Liang Ji-ye, et al. Rough set theory and methods[M]. Beijing: Science Press, 2001 (in Chinese)

张文修, 吴伟志, 梁吉业, 等. 粗糙集理论与方法[M]. 北京: 科学出版社, 2001

[11] Zhang Xian-yong, Xiong Fang, Mo Zhi-wen. Rough Set Model Based on Logical OR Operation of Precision and Grade[J]. Pattern Recognition and Artificial Intelligence, 2009, 17(9): 151-155 (in Chinese)

张贤勇, 熊方, 莫志文. 精度与程度的逻辑或粗糙集模型[J]. 模式识别与人工智能, 2009, 17(9): 151-155

[12] Zhang Xian-yong, Xiong Fang, Mo Zhi-wen. Properties of Approximation Operators of Logical Difference Operation of Grade and Precision[J]. Mathematics in Practice and Theory, 2011, 18(41): 209-213

[13] Zhang Xian-yong, Mo Zhi-wen, Xiong Fang, et al. Comparative study of variable precision rough set model and graded rough set model [J]. International Journal of Approximate Reasoning, 2012, 53: 104-116

[14] Zhang Xian-yong, Miao Duo-qian. Two basic double-quantitative rough set models of precision and grade and their investigation using granular computing[J]. International Journal of Approximate Reasoning, 2013, 54: 1130-1148

(上接第 268 页)

[12] Laura B, Ioannis T, Constantin A. A Comparison of Novel and State-of-the-Art Polynomial Bayesian Network Learning Algorithm[C]// Proceedings of the 20th National Conference on Artificial Intelligence (AAAI). AAAI Press, 2005: 739-745

[13] David M C, Christopher M. Monotone DAG faithfulness: A bad assumption [R]. Microsoft, 2003

[14] Shunkai F, Michel C D, Sein M, et al. A survey on advances in Markov blanket induction algorithms[C]// Proceedings of the

10th ICNC-FSKD. Xiamen, China, 2014

[15] Shunkai F, Michel C D. Tradeoff analysis of different Markov blanket local learning approaches[C]// Proceedings of the 12th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD). Osaka, Japan: Springer, 2008: 562-571

[16] Facundo B, Dimitris M, Vsant H. Efficient Markov network structure discovery using independence tests [J]. Journal of Artificial Intelligence Research, 2009, 35(1): 449-484