

# 学习贝叶斯网络结构的混合粒子群算法

汪春峰<sup>1</sup>, 吕军成<sup>2</sup>

1. 河南师范大学数学与信息科学学院, 河南新乡 453007
2. 郑州工业贸易学校, 郑州 450007

**摘要** 针对直接使用粒子群算法进行结构学习效率较低的缺陷, 基于无约束优化, 提出一种贝叶斯网络结构学习的混合粒子群算法。该算法首先构造并求解一无约束优化问题, 其最优解对应的无向图中的边可为结构学习提供一搜索范围, 缩小粒子群算法的搜索空间, 然后在缩小的空间中完成对贝叶斯网络的结构学习, 从而提高了粒子群算法的学习效率。仿真试验结果表明, 该混合粒子群算法可以快速、准确地学习到最优贝叶斯网络结构。

**关键词** 贝叶斯网络; 结构学习; 无约束优化; 混合粒子群算法

**中图分类号** TP181

**文献标志码** A

**doi** 10.3981/j.issn.1000-7857.2013.22.008

## Hybrid Particle Swarm Algorithm for Learning Bayesian Network Structure

WANG Chunfeng<sup>1</sup>, LÜ Juncheng<sup>2</sup>

1. Department of Mathematics, Henan Normal University, Xinxiang 453007, Henan Province, China
2. Zhengzhou Trade and Industry Schools, Zhengzhou 450007, China

**Abstract** In order to overcome the defects existing in the lower efficiency of structure learning caused by directly applying particle swarm algorithm to it, i.e. the search space is too large; a hybrid particle swarm algorithm for Bayesian network structure learning is presented based on unconstrained optimization problem. Firstly, for the algorithm, an unconstrained optimization problem is established and solved; the edges in the undirected graph corresponding to the optimal solution could provide a search range for structure learning and reduce the search space of particle swarm algorithm; then, Bayesian network structure learning is completed in the reduced space. Therefore, the leaning efficiency of particle swarm algorithm is raised. The simulation results indicate that the proposed method is able to quickly and accurately learn the optimum Bayesian network structure.

**Keywords** Bayesian network; structure learning; unconstrained optimization; particle swarm algorithm

### 0 引言

贝叶斯网络起源于贝叶斯统计学, 是近 20 年发展起来的处理不确定信息和进行概率推理最有力的工具, 是一种对概率关系的有向图解描述。贝叶斯网络可综合专家先验知识和实例数据的分布特征, 调整其网络结构和概率分布, 识别变量之间潜在的关系及关联程度。发展至今, 贝叶斯网络已被广泛应用于人工智能领域, 具体包括故障检测、医疗诊断、交通管理、军事目标自动识别、数据挖掘、作战意图自动估计和信息融合等方面。

学习贝叶斯网络是指以某种测度寻找一种网络, 使其可以较好地与给定实例数据相拟合。就一般意义而言, 寻找一种网络, 包括寻找一种有向无环图 (Directed Acyclic Graph, DAG) 结构和获得与 DAG 中每个结点相关的条件概率表, 前

者称为网络结构学习, 后者称为网络参数学习。结构学习是利用训练样本集, 结合先验知识, 找出合适的贝叶斯网络拓扑结构, 一般需要考虑两个方面, 即模型选择和模型优化<sup>[1]</sup>, 模型选择决定了评判不同模型优劣的准则, 如打分方法和条件独立性测试, 而模型优化则是要把最优模型结构找出来, 如爬山算法。参数学习是在给定贝叶斯网络拓扑结构的情况下, 确定与各个节点相关的条件概率表。在这两块内容中, 由于参数学习可依据结构和数据集来确定, 所以结构学习这块内容是贝叶斯网络学习的核心部分。有效的结构学习方法是构建最优贝叶斯网络结构的基础。

现有贝叶斯网络结构学习的方法大致可归为 3 类。第 1 类是基于依赖关系的学习算法, 其基本思想是利用样本中出现的条件独立关系构建贝叶斯网络结构, 比较经典的算法有

收稿日期: 2013-03-18; 修回日期: 2013-05-26

基金项目: 国家自然科学基金项目 (11171094); 河南师范大学博士科研启动课题 (qd12103); 河南师范大学校级骨干教师培养资助项目; 河南省基础与前沿技术研究计划项目 (132300410285); 河南省教育厅科学技术研究重点项目 (13A11054)

作者简介: 汪春峰, 博士, 副教授, 研究方向为贝叶斯网络学习、最优化理论方法及应用, 电子信箱: wangchunfeng09@126.com

PC 等<sup>[2]</sup>, 此类算法的时间复杂度相对较低, 但对训练数据集的依赖性较大, 如果训练数据量不是充分大或存在误差, 结果通常是不可靠的; 此外, 许多基于依赖分析的算法需要事先知道节点的顺序, 而在真实问题中, 这一条件往往很难满足, 在不要求节点有序的情况下, 基于依赖分析的贝叶斯网络结构学习算法的时间复杂度至少为  $O(n^4)$ <sup>[3]</sup>。第 2 类是基于打分搜索的算法, 其基本思想是利用打分函数寻找最高得分的贝叶斯网络结构, 比较典型的算法有 MMHC<sup>[4]</sup>、ACO<sup>[5]</sup>等, 这类方法在准确性方面比第 1 类方法有所提高, 但当网络节点较多时, 网络结构的搜索空间呈指数级增长, 从而导致算法效率大为降低, 因此, 这类算法只适用于具有一定结构先验知识的局部学习和启发式学习中。第 3 类是混合算法, 这类算法是以上两类算法思想的混合, 比较典型的算法有 I-ACOB<sup>[6]</sup>、模拟退火方法<sup>[7,8]</sup>和粒子群算法<sup>[8-11]</sup>等, 因其能克服前两类算法的缺陷, 目前已成为研究热点之一。本文研究一种结构学习效率较高的混合粒子群算法。

### 1 贝叶斯网络的基本概念

贝叶斯网是一种帮助人们将概率统计应用于复杂领域, 进行不确定性推理和数据分析的工具, 又称信度网络、因果概率网络。贝叶斯网可以表示为一个二元组, 即

$$BN=(G, \theta)$$

其中,  $G=(X, E)$  是一个非循环的有向无环图 (DAG), 包括节点变量集合  $X=(x_1, x_2, \dots, x_n)$  和连接节点的有向边集  $E$ , 它捕捉了结构特性, 降低了推理、决策和学习的复杂度;  $\theta$  是一个条件概率分布集, 其中  $\theta_i \in \theta$  表示贝叶斯网络在给定节点  $v_i$  的父节点时的条件概率。两部分组合在一起定义了唯一的联合概率分布, 贝叶斯网是联合概率分布的一个自然、紧凑的表示形式。

考虑具有  $n$  个属性节点  $(x_1, x_2, \dots, x_n)$  的贝叶斯网络, 假定对于每个节点  $x_i$ , 在给定其父节点集  $Pa(x_i)$  时,  $x_i$  与  $X \setminus Pa(x_i)$  条件独立, 则联合概率  $P(x_1, x_2, \dots, x_n)$  满足下式:

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | Pa(x_i)) \quad (1)$$

式(1)表明贝叶斯网络实质上是联合概率分布  $P(x_1, x_2, \dots, x_n)$  所有条件独立的一个图形化表示。

给定离散变量集  $(x_1, x_2, \dots, x_n)$  上的数据集  $D$ , 贝叶斯网络结构学习是指尽可能结合先验知识, 找到和样本数据拟合最好的网络拓扑结构。通常的做法是定义一个评分函数, 评判某个具体结构所反映的独立关系和样本的匹配程度, 选择适宜的搜索算法搜索分值最高的网络模型, 即

$$G^* = \operatorname{argmax}_{G \in \mathcal{G}_n} f(G; D) \quad (2)$$

式(2)中  $f(G; D)$  是度量网络  $G$  与数据集  $D$  拟合程度的打分函数;  $\mathcal{G}_n$  是由定义在节点集  $X$  上的所有 DAG 构成的集合。

目前存在的打分函数都是在统计数据集合的基础上设计的, 大体上可以分为依据信息论原理设计的打分函数和依

据贝叶斯方法设计的打分函数。这些打分函数通常具有一个很重要的性质, 即可分解性:

$$f(G; D) = \sum_{i=1}^n f(x_i, Pa(x_i); N_{x_i, Pa(x_i)}) \quad (3)$$

式中,  $N_{x_i, Pa(x_i)}$  表示  $x_i$  与  $Pa(x_i)$  在给定数据集  $D$  中的统计量。利用这种可分解性, 当网络中某节点处发生边的添加、删除或反向操作, 需要计算新网络的得分时, 只需计算该节点处的局部分值即可, 因为其他节点处未发生改变。这样可以大大减少计算量, 在设计算法方面具有重要作用。

在诸多打分函数中, 比较常用的有 BIC、AIC 和 BDe 等, 本文采用 BIC 打分函数, 其定义如下:

$$f(G; D) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \left( \frac{N_{ijk}}{N_{ij}} \right) - \frac{1}{2} \log(N) \sum_{i=1}^n (r_i - 1) q_i \quad (4)$$

式中,  $r_i$  表示变量  $x_i$  有  $r_i$  种取值可能;  $q_i$  表示  $x_i$  的父节点  $Pa(x_i)$  有  $q_i$  种取值可能;  $N_{ijk}$  表示样本集合  $D$  中变量  $x_i$  取第  $k$  个值的频数;  $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$  表示样本集合  $D$  中  $Pa(x_i)$  取第  $j$  个值的频数。

### 2 混合粒子群算法

#### 2.1 粒子群算法

粒子群算法最早是在 1995 年由 Kenney 和 Eberhart<sup>[12]</sup>提出的一种群智能优化算法。该算法源于对鸟群捕食行为的研究, 具有程序设计简单、调节参数少等优点。其基本思想是, 将优化问题的每个解看作“粒子”, 每个粒子对应一个由优化函数决定的适应度, 适应度的大小决定了粒子的优劣, 粒子运动的方向和距离由速度决定。在每一次的迭代中, 粒子通过跟踪 2 个“极值”来更新自己, 第一个是粒子本身找到的最优解, 这个解叫做个体极值  $Pbest$ , 另一个是整个种群找到的最优解, 这个解叫做全局极值  $Gbest$ 。

在确定了 2 个最优值  $Gbest$  和  $Pbest$  之后, 粒子可根据式(5)更新自己的速度和位置:

$$\begin{cases} v_{k+1} = \omega v_k + c_1 r_1 (Pbest - x_k) + c_2 r_2 (Gbest - x_k) \\ x_{k+1} = x_k + v_{k+1} \end{cases} \quad (5)$$

式中,  $k$  表示迭代次数;  $v_k$  表示粒子在第  $k$  次迭代时的速度;  $x_k$  表示粒子在第  $k$  次迭代时的位置;  $r_1$  和  $r_2$  为介于  $[0, 1]$  之间的随机数;  $c_1$  和  $c_2$  为加速常数, 在  $[0, 2]$  之间随机选取;  $\omega$  为惯性权重, 用来控制上次迭代的速度对当前速度的影响, 较大的  $\omega$  可以加强粒子群优化算法全局搜索能力, 较小的  $\omega$  可以加强局部搜索能力。

最近, 文献[13]证明了粒子群优化算法的进程与粒子的速度无关, 提出了简化的粒子群算法:

$$x_{k+1} = \omega x_k + c_1 r_1 (Pbest - x_k) + c_2 r_2 (Gbest - x_k)$$

在这个简化式中, 粒子群优化算法不含有速度项, 其中右边

第1项为“历史部分”,表示过去对现在的影响,通过 $\omega$ 调节影响程度;第2项为“认知”部分,表示粒子自身的思考;第3项为“社会”部分,表示粒子间的社会信息共享。

### 2.2 混合粒子群算法及其关键环节

文献[14]提出了一种学习贝叶斯网络结构的粒子群算法,但该方法是在整个贝叶斯网络结构空间中进行学习的,故搜索空间比较大。

本文提出一种混合粒子群算法(HYPSOB),首先需要构造一无约束优化问题,该问题的最优解对应一无向图,无向图中的边可以大大减小粒子群算法搜索最优贝叶斯网络结构的搜索空间,从而改善单纯使用粒子群算法效率比较低的缺陷。

#### 2.2.1 无约束优化问题

令 $\chi_{ijk}^2$ 表示自由度为

$$df=(n_i-1)(n_j-1)n_k$$

的 $\chi^2$ 分布(其中 $n_i$ 是 $x_i$ 的可取值个数), $\chi_{ij,\alpha}^2$ 表示自由度为

$$df=(n_i-1)(n_j-1)n_k$$

显著性水平为 $\alpha$ 的 $\chi^2$ 分布值。

定义1 1-步依赖系数考虑变量 $x_i$ 和 $x_j$ ,称

$$c_{ij,\alpha}^{(1)}=\min_{k \neq i,j} \{\chi_{ijk}^2 - \chi_{ij,\alpha}^2\}$$

为 $x_i$ 与 $x_j$ 之间的1-步依赖系数。

注1:显然,如果 $c_{ij,\alpha}^{(1)} > 0$ ,则无论是否有其它变量加入,变量 $x_i$ 和 $x_j$ 之间都满足统计学意义上以 $(1-\alpha)\%$ 依赖,即它们之间极大可能存在一条边。如果 $c_{ij,\alpha}^{(1)} \leq 0$ ,则它们之间可能是条件独立的。在实际编程实现时,根据文献[15]中的定理1, $c_{ij,\alpha}^{(1)}$ 可用下面一个近似量替代:

$$c_{ij,\alpha}^{(1)} = \min_{k \neq i,j} \{2NMI_D(x_i, x_j | x_k) - \chi_{ij,\alpha}^2\}$$

其中 $N$ 为样本数据条数; $MI_D(x_i, x_j | x_k)$ 为节点变量 $x_i, x_j$ 与条件节点变量 $x_k$ 之间的条件互信息。

例如,考虑一个四节点的贝叶斯网络如图1所示,将图中的Cloudy、Sprinkler、Rain和Wet Grass分别简记为节点C、S、R和W,并假定每个节点变量可以取2个值。为了判断节点S和W之间是否相互依赖,即节点S和W之间是否存在边,我们可以通过计算两变量间的1-步依赖系数 $c_{SW,\alpha}^{(1)}$ 确定。

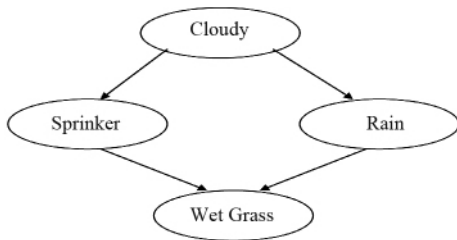


图1 四节点的贝叶斯网络

Fig. 1 Four-node Bayesian network

假定 $\chi^2$ 分布的显著性水平 $\alpha=0.01$ ,此时 $\chi^2$ 分布自由度为

$$df=(2-1)*(2-1)*2=2$$

对应的 $\chi^2$ 分布值为9.2103,从而根据定义1可计算

$$c_{SW,\alpha}^{(1)}:c_{SW,\alpha}^{(1)} = \min\{c_{SWC,\alpha}^{(1)}, c_{SWR,\alpha}^{(1)}\} = 9.2103$$

如果

$$c_{SW,\alpha}^{(1)} > 0$$

则节点S和W之间以99%的概率存在一条边。

定义2 构造矩阵 $C=(c_{ij})$ 如下

$$c_{ij} = \begin{cases} c_{ij,\alpha}^{(1)} & i \neq j \\ 0 & i = j \end{cases}$$

称矩阵 $C$ 为1-步依赖系数矩阵。

注2:注意到 $\chi_{ijk}^2 = \chi_{jik}^2$ ,因此有 $c_{ij,\alpha}^{(1)} = c_{ji,\alpha}^{(1)}$ ,即 $C$ 是一对称矩阵。

定义3 如果存在节点 $x_k$ 使得 $\chi_{ijk}^2 < \chi_{ij,\alpha}^2$ 成立,则称节点 $x_i$ 和 $x_j$ 以显著性水平 $\alpha$ 局部条件独立(LCI)。

根据 $c_{ij,\alpha}^{(1)}$ 以及局部条件独立的定义,有以下结论。

引理1  $x_i$ 和 $x_j$ 以显著性水平 $\alpha$ 局部条件独立(LCI),当且仅当 $c_{ij,\alpha}^{(1)} < 0$ 。

定义4 若对于 $\forall k \neq i, j$ ,都有 $\chi_{ijk}^2 < \chi_{ij,\alpha}^2$ 成立,则称节点 $x_i$ 和 $x_j$ 以显著性水平 $\alpha$ 全局条件独立(GCI)。

一个贝叶斯网络结构可以表示成一个邻接矩阵 $Y=(y_{ij})$ 的形式,其中

$$y_{ij} = \begin{cases} 1 & (x_i, x_j) \in E \\ 0 & (x_i, x_j) \notin E \end{cases}$$

通过使用邻接矩阵 $Y=(y_{ij})$ 和1-步依赖系数,可以构造一个度量网络全局依赖的量。

定义5 对于一个邻接矩阵为 $Y=(y_{ij})$ 的贝叶斯网络,其1-步全局依赖度量为

$$H(y, \alpha) = \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} y_{ij}$$

在定义5的基础上,可构造如下无约束优化问题:

$$\max H(y, \alpha) = \sum_{i=1}^n \sum_{j=1, j \neq i}^n c_{ij} y_{ij} \quad (6)$$

由于系数矩阵是一对称阵,所以由上述无约束优化问题确定出来的邻接矩阵对应的图必是一无向图。再根据 $c_{ij,\alpha}^{(1)}$ 的定义可知,无向图中所保留的边是最优贝叶斯网络结构的一些候选边。

#### 2.2.2 HYPSOB算法关键环节

为了将粒子群算法应用于BNs结构学习,我们需要根据DAG空间的结构特点定义粒子的位置 position 和粒子的速度 velocity,以及以下几种运算规则:

1) move (position, velocity)  $\rightarrow$  position, 即粒子在位置 position 获得速度 velocity 时如何进行移动;

2) subtraction (position, position)  $\rightarrow$  velocity, 即已知两个粒子的位置 position, 如何确定粒子由某一位置移动到另一位置的速度 velocity;

3) add(velocity, velocity, velocity) → velocity, 即如何确定粒子由某一位置移动到下一位置时的速度。

下面逐一进行介绍。

在算法第  $t$  步, 假定粒子  $i$  的当前位置为  $G_{i,t}$ 。粒子在此次迭代之前的最好位置为  $G_{i,t}^p$ , 整个粒子群的当前最好位置为  $G_t^p$ 。

在贝叶斯网络结构学习中, 根据其搜索空间的特点, 一个自然的想法是将粒子的位置定义为一个 DAG。另外, 由前面内容可知, 一个 DAG 可以表示成一个邻接矩阵的形式, 因此粒子的位置具体可表示为其对应有向无环图的邻接矩阵的形式。图 2 所示为四节点的贝叶斯网络结构图。

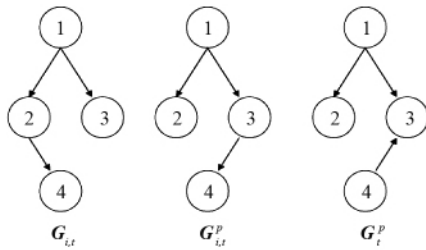


图 2 粒子不同位置对应的 DAG

Fig. 2 Corresponding DAG for particle different positions

根据邻接矩阵的表示形式, 位置  $G_{i,t}$  和  $G_{i,t}^p$  可表示为

$$G_{i,t} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$G_{i,t}^p = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

在粒子群算法中, 因速度的作用在于移动粒子和度量两个粒子位置之间的差异, 所以为了和粒子“位置”的定义一致起来, 粒子的速度同样定义为一个  $n \times n$  阶的矩阵形式。记粒子  $i$  在第  $t$  步的速度为  $v_{i,t}$ , 其第  $i$  行第  $j$  列位置的元素为  $v_{ij} \in \{-1, 0, 1\}$ 。当速度作用于粒子的某一位置时,  $v_{ij} = -1$  意味着 DAG 中节点  $i$  到节点  $j$  之间的边将被移除;  $v_{ij} = 0$  意味着 DAG 中节点  $i$  到节点  $j$  之间的边保持不变;  $v_{ij} = 1$  意味着 DAG 中节点  $i$  到节点  $j$  之间将增加一条边; 在 DAG 中, 因节点  $i$  不能有指向其自身的边, 所以  $v_{ii} = 0$ 。

在以上位置和速度定义的基础上, 下面介绍操作算子 move(position, velocity), subtraction(position, position) 和 add(velocity, velocity, velocity) 的具体实现。

首先看操作算子 move(position, velocity)。当速度  $v$  作用于粒子位置  $G$  时,  $G$  中元素  $g_j$  将根据  $v_j$  进行改变, 其具体改变过程如下:

$$g'_j = \begin{cases} 1 & g_j=0, v_j=1 \text{ 或 } g_j=1, v_j=0 \\ 0 & g_j=0, v_j=0 \text{ 或 } g_j=1, v_j=-1 \\ 1 & g_j=1, v_j=1 \\ 0 & g_j=0, v_j=-1 \end{cases}$$

例如以下速度

$$v = (v_1, v_2, v_3, v_4) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \end{pmatrix}$$

可以将图 2 中所示的粒子  $i$  由位置  $G_{i,t}$  移动到  $G_{i,t}^p$ 。

其次看操作算子 subtraction(position, position)。该算子的作用在于确定两位置之间的差异, 其定义如下:

$$v_{ij} = \begin{cases} 0 & g_j = g'_j \\ 1 & g_j = 1, g'_j = 0 \\ -1 & g_j = 0, g'_j = 1 \end{cases} \quad (7)$$

根据式 (7), 对于图 2 中的  $G_{i,t}$  和  $G_{i,t}^p$ , 由操作算子 subtraction( $G_{i,t}, G_{i,t}^p$ ) 得到的速度  $v$  如下:

$$v = (v_1, v_2, v_3, v_4) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

最后看操作算子 add(velocity, velocity, velocity)。其作用在于通过组合三种运动趋势产生移动粒子的新速度, 假定

$$v_a = [v_{a,1}, v_{a,2}, \dots, v_{a,n}]$$

表示粒子的初始速度,

$$v_b = [v_{b,1}, v_{b,2}, \dots, v_{b,n}]$$

是由操作算子 subtraction( $G_{i,t}^p, G_{i,t}$ ) 得到的速度,

$$v_c = [v_{c,1}, v_{c,2}, \dots, v_{c,n}]$$

是由操作算子 subtraction( $G_t^p, G_{i,t}$ ) 得到的速度, 则操作算子 add( $v_a, v_b, v_c$ ) 的定义如下:

$$v_{new,i} = \begin{cases} v_{a,i} & \text{rand} < pr_a \\ v_{b,i} & \text{rand} < pr_b \\ v_{c,i} & \text{rand} < pr_c \end{cases} \quad (8)$$

其中  $pr_a + pr_b + pr_c = 1$ 。式 (8) 中的  $v_{a,i}, v_{b,i}, v_{c,i}$  分别表示速度矩阵  $v_a, v_b, v_c$  的第  $i$  列。这里会出现一个问题, 即如何选取式 (8) 中出现的概率值  $pr_a, pr_b, pr_c$ 。本文的设置方法是, 在初始阶段, 为了让粒子能够搜索新的区域将  $pr_a$  设置为较大的值, 随着搜索过程的进行,  $pr_b, pr_c$  逐渐增加, 以使算法可以在当前获得的最优位置的邻域内进行搜索。具体取法如下:

$$pr_a = 0.7 - t \times 5 / 200$$

$$pr_b = 0.15 + t \times 0.25 / 200$$

$$pr_c = 0.15 + t \times 0.25 / 200$$

其中  $t$  为迭代次数。

### 2.2.3 HYP SOB 算法流程

下面给出本文学习贝叶斯网络结构的混合粒子群算法的具体过程。

输入: 训练集  $D=\{d^1, d^2, \dots, d^n\}$ , 粒子群规模  $N_p$ , 每个节点的最大父节点数  $u$ , 置信度水平  $\alpha$ , 最大迭代次数  $N_c$ .

/\* 无约束优化问题求解阶段 \*/

依据式(6)构造一无约束优化问题, 求其最优解, 获得一无向图。

/\* 初始粒子群的产生阶段 \*/

以无向图为基础, 随机产生  $N_p$  个粒子群的位置  $G_{i,0}$ , 随机产生  $v_i$ 。以操作算子  $move(G_{i,0}, v_i)$  产生  $G_{i,0}^p$  (确保  $G_{i,0}^p$  是有向无环图), 从中确定出当前粒子群的最优位置  $G_i^g$ 。

/\* 主循环阶段 \*/

$t=1$

当  $t < N_c$  时

对每个粒子  $i$ , 随机产生  $v_a$

置  $v_b = subtraction(G_{i,t}^p, G_{i,t}^g)$

置  $v_c = subtraction(G_i^g, G_{i,t}^g)$

置  $v_{new} = add(v_a, v_b, v_c)$

$G_{i,t+1} = move(G_{i,t}, v_{new})$

$G_{i,t+1} = repairbyMutualInfo(G_{i,t+1}, D)$

$G_{i,t+1} = repairMaxparents(G_{i,t+1}, u)$

若  $score(G_{i,t+1}) > score(G_{i,t}^p)$

则置  $G_{i,t+1}^p = G_{i,t+1}$

若  $score(G_{i,t+1}) > score(G_i^g)$

则置  $G_i^g = G_{i,t+1}$

$t=t+1$

循环终止

输出: 最优贝叶斯网络结构  $G_i^g$

HYP SOB 算法中有两个子程序  $repairbyMutualInfo(\cdot)$  和  $repairMaxparents(\cdot)$ , 其作用分别为: 当产生的新网络结构中含有环时, 程序  $repairbyMutualInfo(\cdot)$  将计算环节点间的互信息, 并删除环中具有最小互信息节点之间的边, 实现去环操作; 当产生的新的有向无环图中某节点的父节点个数超过了最大限  $u$  时, 程序  $repairMaxparents(\cdot)$  会从父节点中选择不超过最大上限的最好子集作为父节点集。

### 3 仿真试验

为了测试本文方法的性能, 我们把本文所提出的 HYP SOB 算法与直接使用粒子群算法学习贝叶斯网络结构的 PSOB 算法进行了试验比较。选取图 3 所示的 Asia 网络作为学习对象, 图中的 VA、S、T、LC、B、TOC、D、X 分别表示 Visit to Asia、Smoking、Tuberculosis、LungCancer、Bronchitis、TBor Cancer、Dys、Xray。

程序实现采用 Matlab 7.0, 数值实验平台为 Pentium 4、3.06GHZ CPU、512M 的微机。在算法中, 参数设置为: 显著性水平  $\alpha=0.5\%$ , 粒子群规模为 30, 最大迭代次数为 60。

经过仿真对比发现, 对于本文提出的 HYP SOB 算法, 在

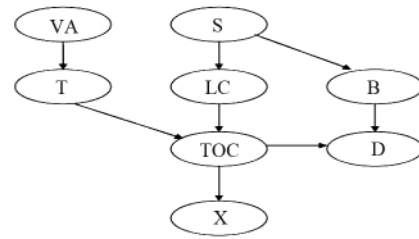


图 3 真实的 Asia 网络

Fig. 3 True Asia network

样本数据条数为 800 时即可学习到比较接近真实 Asia 网络的网络结构, 如图 4 所示。然而对于直接使用粒子群进行贝叶斯网络结构学习的 PSOB 算法, 要获得同样好的结果, 则至少需要 3000 条样本数据。

在样本量为 15000, 粒子群规模为 150, 最大迭代次数为

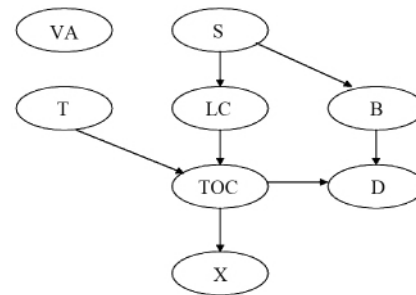


图 4 学习到的贝叶斯网络

Fig. 4 Learned Bayesian network

30 时, 本文方法学习得到了真实的 Asia 网络。图 5 为本文方法学习得到真实 Asia 网络时迭代次数与对应最优得分之间的关系。

由此可见, 本文方法在小样本数据集下即可学习到比较

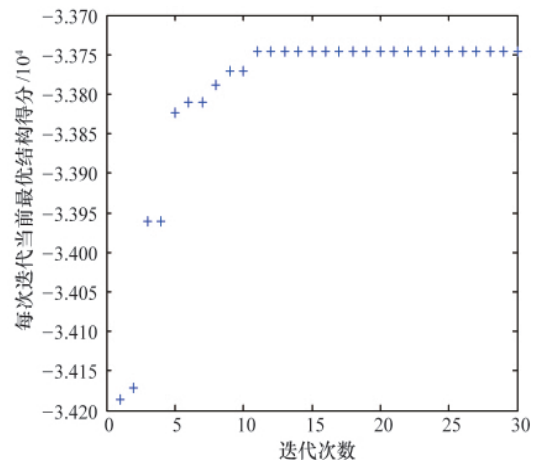


图 5 学习到真实网络结构时迭代次数与最优分值之间的关系

Fig. 5 Relation between iteration times and current optimal scores with learned true network structure

好的网络结构,在样本量足够大时完全可以正确地学习到真实的网络结构。

#### 4 结语

本文首先通过定义 1-步依赖系数,引入 1-步全局依赖度量等概念,构造了一个无约束优化问题。该无约束优化问题最优解对应的无向图中的边,限制了粒子群优化算法搜索贝叶斯网络结构的空问。在此基础上,结合粒子群算法,提出一种效率较高的学习贝叶斯网络结构的混合粒子群算法。仿真试验显示,本文方法在小样本数下即可获得比较接近真实网络的网络结构,证明了本方法的可行性。

#### 参考文献 (References)

- [1] 张连文,郭海鹏. 贝叶斯网引论[M]. 北京: 科学出版社, 2006.  
Zhang Lianwen, Guo Haipeng. Introduction to Bayesian networks [M]. Beijing: Science Press, 2006.
- [2] Cheng Jie, Greiner Russell, Kelly Jonathan, et al. Learning Bayesian networks from data: an information theory based approach[J]. Artificial Intelligence, 2002, 37(2): 43-90.
- [3] Peng Hanchuan, Ding Chris. Structure search and stability enhancement of Bayesian networks [C]// Proceedings of 3rd IEEE International Conference on Data Mining, Melbourne, AUS: [s.n.], 2003, 621-624.
- [4] Tsamardinos Ioannis, Brown Laura, Aliferis Constantin. The max-min hill-climbing Bayesian network structure learning algorithm[J]. Machine Learning, 2006, 65(1): 31-78.
- [5] de Campos Luis M, Fernández-Luna Juan M, Gámez José A, et al. Ant colony optimization for learning Bayesian networks [J]. International Journal of Approximate Reasoning, 2002, 31(3): 291-311.
- [6] Ji Junzhong, Zhang Hongxun, Hu Renbing. Bayesian network learning algorithm based on independence test and ant colony optimization [J].

- Acta Automatica Sinica, 2009, 35: 281-288.
- [7] 汪春峰,王艳玲. 贝叶斯网络结构学习的限制型模拟退火方法[J]. 河南师范大学学报: 自然科学版, 2013, 41: 6-9.  
Wang Chunfeng, Wang Yanling. Journal of Henan Normal University: Natural Science Edition, 2013, 41: 6-9.
- [8] Martin J, Jan N. A simulated annealing-based method for learning Bayesian networks from statistical data[J]. International Journal of Intelligent Systems, 2006, 21: 335-348.
- [9] Wang Tong, Yang Jie. A heuristic method for learning Bayesian networks using discrete particle swarm optimization[J]. Knowledge and Information Systems, 2010, 24: 269-281.
- [10] Senthil A M, Rao M V C, Chandramohan A. A new and improved version of particle swarm optimization with global-local best parameters [J]. Knowledge and Information Systems, 2008, 16(3): 331-357.
- [11] Heng Xingchen, Qin Zheng, Wang Xianhui, et al. Research on learning Bayesian networks by particle swarm optimization [J]. Information Technology Journal, 2006(5): 540-545.
- [12] Kennedy J, Eberhart R. Particle swarm optimization [C]// Proceedings of IEEE International Conference on Neural Networks. Perth, WA, Australia: [s.n.], 1995: 1942-1948.
- [13] 胡旺,李志蜀. 一种更简化而高效的粒子群优化算法[J]. 软件学报, 2007, 18(4): 861-868.  
Hu Wang, Li Zhishu. A simpler and more effective particle swarm optimization algorithm [J]. Journal of Software, 2007, 18(4): 861-868.
- [14] Du Tao, Zhang Shensheng, Wang Zongjiang. Learning Bayesian networks from data by particle swarm optimization [J]. Journal of Shanghai Jiaotong University: Natural Science Edition, 2006, 11 (4): 423-429.
- [15] de Campos Luis M. A scoring function for learning Bayesian networks based on mutual information and conditional independence tests [J]. Journal of Machine Learning Research, 2006, 7(12): 2149-2187.

(责任编辑 韩黎明)

#### ·学术动态·



### 第 11 届全国博士生学术年会征文

中国科协第 8 届常委会青年工作专门委员会、国务院学位委员会办公室、中国科协组织人事部将于 2013 年 10 月在成都举办第 11 届全国博士生学术年会。

第 11 届全国博士生学术年会设 4 个交流专题。

(1) 信息技术与安全: 大数据; 云计算; 信息网络安全; 信息技术与处理。

(2) 节能环保与污染防治技术: 新能源开发; 节能环保产业理论与实践; 节能减排、废物的资源化和综合利用等新技术的研究与开发; 污染防治技术。

(3) 民用航空科技及产业发展: 通用航空; 支线运输与支线机场建设; 绿色航空; 民航科技发展; 临空经济。

(4) 生物医药: 中医药和天然药物研究; 药物化学研究; 药物分析学研究; 药理学研究; 药剂学研究; 其他与药学相关研究。

欢迎高校、科研院所的在读博士生围绕交流专题积极研究、投稿及参会。

详见中国科协 <http://www.cast.org.cn/n35081/n35096/n13118139/14869886.html>。