

A Novel Hybrid Method for Learning Bayesian Network

Wang Chun-Feng*, Liu Kui

Department of Mathematics, Henan Normal University, Xinxiang, 453007, PR China

* Corresponding author. Tel: +86 13598678640; email: wangchunfeng10@126.com

Manuscript submitted February 2, 2014; accepted March 9, 2015.

doi: 10.17706/jcp.10.2.130-137

Abstract: This paper presents a new hybrid approach for learning Bayesian networks (BNs) based on artificial bee colony algorithm and particle swarm optimization. Firstly, an unconstrained optimization problem is established, which can provide a smaller search space. Secondly, the definition and encoding of the basic mathematical elements of our algorithm are given, and the basic operations are designed, which provide guarantee of convergence. Thirdly, from a known original Bayesian network with probabilistic logic sampling, full samples for the training set and testing set are generated, and then the structure of Bayesian network is learned from complete training set by using our method. The simulation experimental results show that our method is effective.

Key words: Bayesian network structure learning, ABC, PSO, unconstrained optimization.

1. Introduction

With the community of artificial intelligence, Bayesian networks(BN) is an important model, and also a powerful formalism to model the uncertainty knowledge in practice [1]. Since the standard method to construct BN is a labor-intensive process of eliciting knowledge from experts, the approach to learn BN from data has become an increasingly active area of research. Unfortunately, high computational requirements and the lack of power [2]-[6].

The learning task of the Bayesian networks can be decomposed into two subtasks which are the learning of graph structure and parameters. Structure learning is to identify the topology of the network, and parameter learning is to determine the conditional probabilities for a given network topology. This paper focus on structure learning.

From the review of solving mechanism, these methods for learning BN can be classified into two categories: the dependency analysis approaches and the score-and-search approaches [7]-[10]. The first category is based on constraints, which poses the learning process as a constraint satisfaction problem, and then constructs a network structure by testing the conditional independence relations. The second approach poses the learning problem as a structure optimization problem, i.e. it uses a score metric to evaluate every candidate network structure, and then finds a network structure with the best score. The score used to evaluate the network could be derived from information theory, Bayesian statistics and MDL principle. Though the implement of the former approach is relatively simple, the computations for higher-order testings are complex and irresponsible. Moreover, the precision of learning a model is hard to ensure, thus the score-and-search approach gradually becomes a popular approach for learning Bayesian network. The algorithm proposed in this paper belongs to the score-and-search approach.

Since searching the best network structure is an NP-hard problem, the problem is usually approached by

using heuristic methods, such as greedy hill-climbing, simulated annealing, genetic algorithm and PSO algorithm [11], [12]. Among these heuristic methods, PSO have been intensively researched and proved being effective in learning Bayesian network. However, PSO algorithm has two drawbacks in learning Bayesian network which are its expensive computational cost and premature convergence. Recently, ABC algorithm has been successfully applied in many areas [13]. Karaboga and Basturk used ABC algorithm to optimize a large set of numerical test function. Comparing with PSO algorithm, differential evolution algorithm and evolution strategies, ABC's advantages not only lie in its easy implementation and few parameters to adjust, but also lie in its strong search capability in the problem space and the ability of quick discovery of optimal solution. Though ABC algorithm has many advantages, there exists fewer work use it to learn Bayesian network. By our knowledge, the paper [14] is the only paper to learn Bayesian network by using ABC algorithm.

In this paper, we propose a new hybrid method for learning Bayesian network based on PSO and ABC algorithms. In this algorithm, the mechanism of PSO, such as the movement of particle etc., will be used by employed bee, onlooker bee and scout bee to find better food source. In a number of experiments, computational results on benchmark datasets demonstrate the effectiveness of our approach.

The rest of the paper is organized as follows. The concept of BNs and structural learning is introduced in Section 2. Then, we give a brief introduction on ABC algorithm, and discuss the detail of proposed hybrid algorithm in Section 3. In Section 4, some numerical results are reported to show the efficiency of our algorithm. Finally, we conclude the paper in Section 5.

2. Bayesian Network Structure Learning

2.1. Bayesian Networks

Bayesian network (BN) is a directed acyclic graph (DAG), which can be denoted as a triple group (X, A, θ) , where $\langle X, A \rangle$ defines a directed acyclic graph structure G : X is the set of nodes, $x_i \in X$ represents a random variable in a special domain; A is a set of directed arcs, $a_{ij} \in A$ describes a direct probability distribution of x_i given the parent set Π_{x_i} of the variable x_i . $\theta_i = P(x_i | \Pi_{x_i})$ is the conditional probability of node i given the state of its parents Π_{x_i} . As the graph structure G qualitatively characterizes the independence relationships among random variables, and these conditional probability distributions quantify the strength of dependencies between each node and its parents nodes, thus, the joint probability can be written as follows:

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

Given dataset D , the objective of structure learning is to identify the best network structure G that best matches D . For search-and-scoring approaches, how goodness the structure matches dataset is measured by adopted scoring metric. The most common scoring approach to evaluate structure is by the posterior probability of the structure given the data. The posterior probability of a candidate Bayesian network structure BN can be obtained by applying Bayes rules:

$$P(B_s | D) = \frac{P(D | B_s)P(B_s)}{P(D)} \quad (2)$$

where $P(D | B_s)$ is the likelihood of the data given the BN structure, and $P(B_s)$ is the prior probability of the structure BN, and $P(D)$ is the probability of the observed data D . Since $P(D)$ is not dependent on the structure, it can be ignored when trying to find the best scoring function.

The problem is now reduced to find the structure with the maximum likelihood $P(D|B_s)$. In other word, given a structure, these structure are evaluated according to how probable it is that the data are generated from the structure. Because of the decomposition characteristic of Bayesian network as shown in (1), scoring metric commonly used such as Bayesian score, BIC score and MDL score could be decomposed into summation of sunscore of each variable x_i given the states of their parents \prod_{x_i} :

$$Score(G) = \sum_{x_i \in X} Score(x_i | \prod_{x_i}) \tag{3}$$

For example, k_2 score is defined as follows:

$$Score(G) = \sum_{i=1}^n \sum_{j=1}^{q_i} (\log(\frac{(r_i - 1)!}{(N_{ij} - r_i - 1)!}) + \sum_{k=1}^{r_i} \log(N_{ijk}!)) \tag{4}$$

where r_i denotes each variable x_i has r_i possible value assignments; q_i is the number of possible configurations for the variables in \prod_{x_i} relative to D ; N_{ijk} is the number of cases in D where x_i and \prod_{x_i} are instantiated to the j -th configuration. BIC score is give as follows:

$$Score(G) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}} - \sum_{i=1}^n \frac{q_i(r_i - 1)}{2} \log N \tag{5}$$

If a score metric follows (3), it is decomposable. For any decomposable score metrics, it is easy to see that, the best network structure G_{best} could be obtained by searching the best conbiantion of parent \prod_{x_i} for every variable x_i .

In this paper, the BIC score will be used to benchmark the mete-heuristic optimization strategies.

3. Hybrid Algorithm for BN Structure Learning

This section gives a hybrid algorithm based on the ABC algorithm and PSO algorithm. In this algorithm, employed bee, onlooker bee and scout bee use the way of PSO to determine food source. Since the advantages of two algorithms are used, the new algorithm has a very good characteristics, i.e. it can find the optimal Bayesian network structure quickly. In the following, we give the details of our algorithm.

3.1. Classical ABC Algorithm

In ABC algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source denotes the quality (fitness) of the associated solution. The number of the employed bees and the onlooker bees is equal to the number of solutions. At the begin, ABC algorithm generates the initial population of SN solutions (food source positions) randomly. Suppose that each solution $x_i (i=1,2,\dots,SN)$ is a n -dimensional vector. Here, n is the number of optimization parameters. After initialization, the population of the positions (solutions) is subject to repeated cycles of the search processes of the employed bees, the onlooker bees and the scout bees. An employed bee produces a modification on the position (solution) in her memory depending on the local information (visual information) and tests the nectar amount (fitness value) of the new source (new solution). If the nectar amount of the new one is higher than that of the previous one, the bee memorizes the new position and forgets the old one. Otherwise she keeps the position of the previous one in her memory. After all employed bees complete the search process, they share the nectar information of the food sources and their position information with the onlooker bees. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a

food source with a probability related to its nectar amount. As in the case of the employed bee, she produces a modification on the position in her memory and checks the nectar amount of the candidate source. If the nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one. The main steps of the algorithm are given as below:

Step 1. Initialize Population

Step 2. While stop criterion is not met do

Step 3. Place the employed bees on their food sources

Step 4. Place the onlooker bees on the food sources depending on their nectar amounts

Step 5. Send the scouts to the search area for discovering new food sources

Step 6. Memorize the best food source found so far

Step 7. End while

In ABC algorithm, each cycle of the search consists of three steps: sending the employed bees onto their food sources and evaluating their nectar amounts; after sharing the nectar information of food sources, the selection of food source regions by the onlookers and evaluating the nectar amount of the food sources; determining the scout bees and then sending them randomly onto possible new food sources.

3.2. The Mechanism of PSO

In our algorithm, for determining food sources for employed bees, onlooker bees and scout bees, the mechanism of PSO will be adopted. The details of the mechanism are given below. to generate better initial food sources, we first construct an unconstrained optimization problem, its optimal solution can be used to reduce the searching space. The detail is explained below.

As we all known, a Bayesian network can be represented by an adjacency matrix, so, in this paper, adjacency matrix will be used to represent food source (solution or candidate Bayesian network) in ABC algorithm. For example, the adjacency matrix of the Bayesian network in Fig. 1. is

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

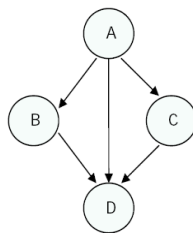


Fig. 1. Four nodes network.

Thus, for consistence with the definition of "food source" of ABC algorithm, the position of a particle in PSO also be represented by a matrix.

Suppose that at step t , the particle i is denoted by $G_{i,t}$ in this paper, the best previous particle i denoted by $G_{i,t}^p$, and the best previous position among particle swarm is denoted by G_t^g .

In PSO algorithm, the velocity is used to move the particle and measure the difference between two positions. In order to convenient operation, the velocity of a particle is also defined as an $n \times n$ matrix. Here the velocity matrix for particle i at step t is denoted by $v_{i,t}$. The element of velocity matrix in row i and column j is denoted by v_{ij} , where $v_{ij} \in \{-1,0,1\}$. when the velocity is applied to a position,

$v_{ij} = -1$ means that the edge from node i to node j in the DAG would be removed, $v_{ij} = 0$ means that the corresponding edge would remain unchanged, and $v_{ij} = 1$ means that the edge from node i to node j would be added. Note that in DAG, an edge from node to itself is forbidden, thus $v_{ii} = 0$.

With the definitions of "position" and "velocity" stated above, we give the other operations: $move(position, velocity)$ and $subtraction(position, position)$.

When a position G_i is moved to G'_i by a velocity v_i , the element g_{ij} is changed according to v_{ij} .

The definition of the operation $move(G, v) \rightarrow G'$ is given:

$$g'_{ij} = \begin{cases} 1, & g_{ij} = 0, v_{ij} = 1 \text{ or } g_{ij} = 1, v_{ij} = 0, \\ 0, & g_{ij} = 0, v_{ij} = 0 \text{ or } g_{ij} = 1, v_{ij} = -1, \\ 1, & g_{ij} = 1, v_{ij} = 1, \\ 0, & g_{ij} = 0, v_{ij} = -1. \end{cases}$$

If $g_{ij} = 0$, when $v_{ij} = 1$, then $g'_{ij} = 1$, which means that the edge from node i to node j would be added; otherwise, $g'_{ij} = g_{ij} = 0$. If $g_{ij} = 1, v_{ij} = -1$ would make this edge to be removed, and $v_{ij} = 0$ or 1 would make $g'_{ij} = g_{ij} = 1$.

In addition, in our algorithm, an operation $add(v_a, v_b, v_c) \rightarrow v_{new}$ is used to combine the three tendencies for moving a particle. Here v_a is original velocity of a particle. v_b is velocity resulted from $subtraction(G_{i,t}^p, G_{i,t})$ by which the particle would move toward its best previous position. v_c is resulted from $subtraction(G_i^g, G_{i,t})$ by which the particle would move to global best previous position among the particle swarm.

By the decomposable score metric, a network G is an agglomeration of the parents vector $g_i(i=1, \dots, n)$. For evaluating Bayesian network structures, a little change in $g_i(i=1, \dots, n)$ may cause $score(g_i)$ changed greatly. When combining v_a , v_b and v_c the velocity column vector v_i should be taken as the basic unit to keep better structure features of $G_{i,t}^p$ and G_i^g .

Suppose $v_a = [v_{a,1}, v_{a,2}, \dots, v_{a,n}]$, $v_b = [v_{b,1}, v_{b,2}, \dots, v_{b,n}]$, $v_c = [v_{c,1}, v_{c,2}, \dots, v_{c,n}]$, and $v_{new} = [v_{new,1}, v_{new,2}, \dots, v_{new,n}]$, the operation $add(v_a, v_b, v_c) \rightarrow v_{new}$ is defined

$$v_{new,i} = \begin{cases} v_{a,i}, & rand < pr_a, \\ v_{b,i}, & rand < pr_b, \\ v_{c,i}, & rand < pr_c, \end{cases} \quad (6)$$

subject to $pr_a + pr_b + pr_c = 1$.

It brings a problem that how to assign the values to the probabilities reasonably which are shown in (6). An empirical heuristic is taken in this paper. In initial phase, pr_a could be set a larger value to make particles moving to search new area. And during the searching, pr_b and pr_c should be increased such that the neighbor areas of best previous positions could be searched more intensively.

3.3. Algorithm Statements

Given the mathematical objects and operations defined above, we propose the hybrid algorithm, which is named after "ABC-PSO-BN". In this algorithm, for improving the convergence, we construct and solve an unconstrained optimization problem by adopting the method of paper [15], and obtain an undirect graph. Based on this undirect graph, the initial food source are generated randomly. This process can reduce the search space of our algorithm greatly. The pseudo code of ABC-PSO-BN is shown as follows:

Algorithm ABC-PSO-BN

Input:

Training data $D = \{x^1, x^2, \dots, x^N\}$, N is the number of data case; the initial population size is S , food sources, employee bees and onlooker bees are $\frac{S}{2}$; u is the upper bound of the number of parents for each node; the max iteration number is $maxcycle$; significal level α ; status l_i for each food source.

Step 1. Solve an unconstrained optimization problem, and obtain an undirect graph by adopting the method of [16]. After that, generate the initial food sources randomly $X_i (i=1, \dots, \frac{S}{2})$.

Step 2. For $c=1$ to $maxcycle$ do

/* employee bee phase*/

For each employed bee, initialize position $G_{i,0}, G_{i,0}^p = G_{i,0}, G_0^g = maxscore(G_{i,0}), i=1, \dots, \frac{S}{2}$.

Generate v_a at random, $G_{i,0} = G_{i,0} + v_a$. Set $v_b = subtraction(G_{i,t}^p, G_{i,t})$, $v_c = subtraction(G_t^g, G_{i,t})$,

$v_{new} = add(v_a, v_b, v_c)$.

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

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

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

If $score(G_{i,t+1}) > score(G_{i,t}^p)$, then $G_{i,t+1}^p = G_{i,t+1}$, end if

If $score(G_{i,t+1}) > score(G_t^g)$, then $G_{i,t+1}^g = G_{i,t+1}$, end if

end for

/* onlooker bee phase*/

for $i=1, \dots, \frac{S}{2}$ do

 Compute probability p_i for each food source X_i based on its fitness value.

end for

Set $q=0, i=1$.

Repeat

if $rand < p_i$ then

 set $q=q+1$.

 for food source X_i , generate the candidate food \bar{X}_i by the way of employee bee, then select the better food source between

X_i and \bar{X}_i .

 end if

$i = i + 1$

if $i = \frac{S}{2}$

 set $i=1$;

 end if

until $q = \frac{S}{2}$

/* scout bee phase*/

if $max(l_i) > limit$ then

 Scout bee will generate a new food source to substitute X_i .

 end if

end for

4. Experimental Evaluation

In order to test the behavior of the method proposed in this paper, we present the experimental results carried out with our algorithm on a standard network data sets (Asia Network). A comparative study with ABC-PSO-BN algorithm and PSO-BN([15]) is also performed. The experiments platform was a personal computer with Pentium 4, 3.06GHZ CPU, 512 M memory, and Windows XP. The algorithm was implemented by Matlab 7.0. In our method, the unconstrained optimization problem was solved by calling the function `bintprog(•)` in Matlab. These experimental parameters were set as follows: the confidence value of χ^2 test is 95%, $S=30$, $limit=10$.

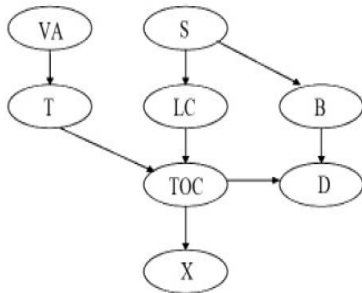


Fig. 2. True Asia network.

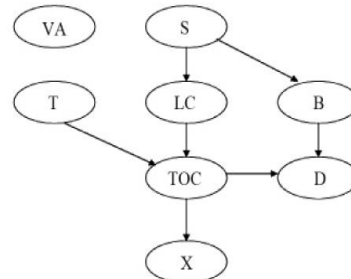


Fig. 3. Better Bayesian network learned by our method.

For Asia network, under dataset is 800, population size is 30, $maxcycle=30$, our method can find better network(see Fig. 3); however the method [15] needs the datasize is 800, population size is 30, $maxcycle=60$. Moreover, with a sample size of 12 000, $maxcycle=20$ and population size of 150, our method finds the best network structure(the relation between iter and score is given in Fig. 4); however, the method [15] needs the datasize is 15 000, $maxcycle=30$ and population size of 150. From the comparision, it can be seen that our method is more efficiency.

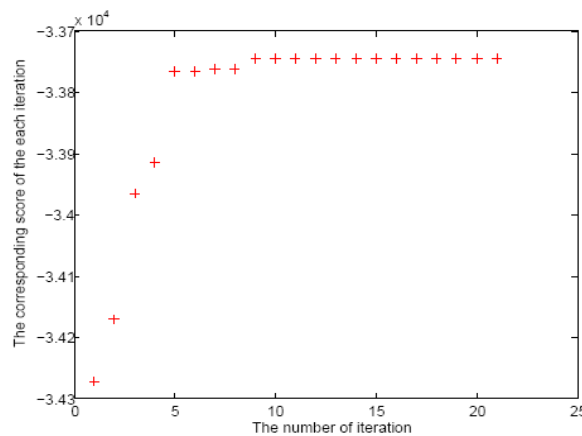


Fig. 4. Relation of iteration and score.

5. Conclusions

We have developed a new hybrid algorithm for learning Bayesian network based on ABC and PSO algorithms. Our algorithm first solved the unconstrained optimization problem to obtain an undirected graph. This procedure can effectively restrict the space of candidate solutions, so that many unnecessary searches can be avoided. The algorithm was tested on a standard network structure. The experimental results illustrate that the new algorithm is superior both in terms of quality of the solutions and computational time.

References

- [1] Heckerman, D., Geiger, D., *et al.* (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20,197-243.
- [2] Ji, J. Z., Hu, R. B., Zhang, H. X., *et al.* (2011). A hybrid method for learning Bayesian networks based on ant colony optimization. *Applied Soft Computing*, 11(4), 3373-3384.
- [3] Pinto, P. C., Nagele, A., Dejon, M., *et al.* (2009). Using a local discovery ant algorithm for Bayesian network structure learning. *IEEE Transactions on Evolutionary Computation*, 13(4), 767-778.
- [4] Ji, J. Z., Hu, R. B., & Liu, C. N. (2009). A Bayesian network learning algorithm based on independence test and ant colony optimization. *Acta. Autom. Sinica.*, 35(3), 281-288.
- [5] Luis, M. C., Juan, M. F., Jose, A. G., *et al.* (2002). Ant colony optimization for learning Bayesian networks. *International Journal of Approximate Reasoning*, 31, 291-311.
- [6] Cheng, J., Greiner, R., Kelly, J., *et al.* (2002). Learning brief networks from data: an information theory based approach. *Artificial Intelligence*, 137, 43-90.
- [7] De Campos, L. M., & Huete, J. F. (2000). A new approach for learning belief networks using independence criteria. *International Journal of Approximate Reasoning*, 24(1), 11-37.
- [8] De Campos, L. M., Fernandez-Luna, J. M., Ganez, J. A., *et al.* (2002). Ant colony optimization for learning Bayesian networks. *International Journal of Approximate Reasoning*, 31(3), 291-311.
- [9] Larranaga, P., Poza, M., Yurramendi, Y., *et al.* (1996). Structure learning of Bayesian network by genetic algorithms: a performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9), 912-926.
- [10] Jose, A. G., & Jose, M. P. (2002). Searching for the best elimination sequence in Bayesian networks by using ant colony optimization. *Pattern Recognition Letters*, 23(1-3), 261-277.
- [11] Man, L. W., & Kwong, S. L. (2004). An efficient data mining method for learning Bayesian networks using an evolutionary algorithm-based hybrid approach. *IEEE Transactions on Evolutionary Computation*, 8(4), 378-404.
- [12] Wang, T., & Yang, J. (2010). A heuristic method for learning Bayesian networks using discrete particle swarm optimization. *Knowledge and Information Systems*, 24, 269-281.
- [13] Karaboga, D., & Basturk, B. (2010). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 24, 269-281.
- [14] Ji, J. Z., Wei, H. K., & Liu, C. N. (2013). An artificial bee colony algorithm for learning Bayesian networks. *Soft Computing*, 17(6), 983-994.
- [15] Wang, C. F., & Lv, J. C. (2013). Hybrid particle swarm algorithm for learning Bayesian network structure. *Science and Technology Review*, 31(22), 50-55.



Wang Chun-Feng was born in 1978. He received his M.S. degree in 2006 from Henan Normal University. He received his Ph.D. degree in 2012 in Xidian University.

His research interests include optimization theory and its applications, Bayesian network learning.



Liu Kui was born in 1978 in Jiaozuo, Henan Province, China, in 1980. He received his B.A. degree in mathematics and applied mathematics from Henan Normal University, Xinxiang, China, in 2004, the M.S. degree in applied mathematics from Xidian University, Xi'an, China, in 2008, and the Ph.D. degree in applied mathematics from Xidian University, Xi'an, China.

His research interests include wireless sensor networks optimization and dynamic networking technology.