A Hybrid Optimization Algorithm for Bayesian Network Structure Learning Based on Database

Junyi Li
Department of Computer Engineering, Dongguan Polytechnic, Dongguan, China
Email: lijunyi68@126.com

Jingyu Chen
School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China

***Abstract*—The process of learning Bayesian networks includes structure learning and parameters learning. During the process, learning the structure of Bayesian networks based on a large database is a NP hard problem. The paper presents a new hybrid algorithm by integrating the algorithms of MMPC (max-min parents and children), PSO (particle swarm optimization) and GA (genetic algorithm) effectively. In the new algorithm, the framework of the undirected network is firstly constructed by MMPC, and then PSO and GA are applied in score-search. With the strong global optimization of PSO and the favorable parallel computing capability of GA, the search space is repaired efficiently and the direction of edges in the network is determined. The proposed algorithm is compared with conventional PSO and GA algorithms. Experimental results show that the proposed algorithm is most effective in terms of convergence speed.**

***Index Terms*—Bayesian network, particle swarm optimization, genetic algorithm, crossover, mutation**

I. Introduction

Bayesian network is an attractive data mining method because of its special characteristics such as expression of uncertain knowledge, capacity of complex probability calculation, and incremental learning from comprehensive priori knowledge, etc. However, Bayesian network always faces an inevitable problem that the network structure is unknown initially. The process of learning Bayesian network includes structure learning and parameters learning. Within the process, learning the structure of Bayesian network based on large database is a NP hard problem. There are various algorithms for learning Bayesian networks. In 1992, Cooper proposed a popular heuristic algorithm called K2 algorithm[1], which is a popularly used approach. But this algorithm is easy to result in local optimum. In 2000, Spirtes proposed an idea of constraint-based approaches for learning Bayesian network. Consequently, an algorithm using local discovery called Max-Min Parents and Children (MMPC) is introduced by Tsamardinos in 2003. Based on the concept of MMPC, Laura E. Brown presented the max-min hill-climbing (MMHC) algorithm in 2005[2]. The approach of MMHC and MMPC is to minimize the searching complexity by reducing the search space, while the scoring algorithm will not revise this error because of the constraint of the search space. Therefore, the results of these two algorithms are prone to local optimum.

To cope with the above problems, we design a hybrid optimization algorithm for Bayesian network structure learning from large database. Firstly, we apply max-min parents and children (MMPC) to construct the framework of the undirected network, and then use particle swarm optimization (PSO) and genetic algorithm (GA) for score-search, which can repair the search space and determine the direction of edges in the network. By study of the respective characteristics and the similarity of PSO and GA, the hybrid algorithm is to search the optimal local and global particles by using the operation of crossover and mutation in GA, which can reduce the randomicity while searching optimal particles in PSO. Finally the searching efficiency can be increased and the accurate structure of Bayesian networks is determined.

II. About Bayesian Network Structure Learning

Bayesian network is shown as a directed acyclic graph $B=(V, E)$. The variables from database are denoted by nodes set $V=\{ x1, x2,..., xn\}$. The directed edges set E reflects the directed relations among variables. The conditional probability exists for each node. For each node $xi \in V$, there exists its father nodes $P(xi|Pa(xi))$. The relation strength between variables is evaluated by the probability between the father and son nodes. This method can accurately determine the relation among multiple variables, and reflect the uncertainty of information in the probabilities.

The process of constructing Bayesian networks includes structure learning and parameters learning. The problem of Bayesian network structure learning can be described as following. Let $x1,x2,...,xn$ be a group of random variables. Let $D=(D1,D2,...,Dn)$ be the corresponding training data set. The task is to find out the best data model S matching the training data set. We can conclude the posterior probability of S by Bayesian theorem as equation (1).

$$P(S \mid D) = \frac{P(D \mid S)P(S)}{P(D)} \quad (1)$$

doi:10.4304/jcp.9.12.2787-2791

In equation (1), *S* represents the probably correct network structure, *P(D/S)* stands for the marginal likelihood, *P(D)* is a regular constant unrelated to the structure. Therefore, as long as the marginal likelihood for each probable structure is calculated, the posterior distribution of the network structure can be determined. Upon the assumption that all the prior probabilities of parameters are in Dirichle distribution[1], the marginal likelihood can be calculated as Equation (2).

$$P(D|S)=\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})} \quad (2)$$

In equation (2), N_{ijk} represents the number of data samples which matching $x_i=k$, $Pa(x_i)=j$. Each variable X_i can be substituted in this equation for the result. This problem is a NP hard problem.

III. Hybrid Optimization Algorithm For Bayesian Network Structure Learning

A. MMPC

MMPC is a kind of local causal discovery algorithm. It is used for working out the father and son nodes collection of variables. Given the data set *D* and the target variable T, MMPC can work out the candidate parents and children (CPC) for every node. The first step of this algorithm is to move variables into the *CPC(T)* for the target node T one by one, using the Max-min strategy heuristic approach. The second step is to remove the undesired variables from *CPC(T)*. The algorithm is detailed as below[2].

MMPC algorithm

Input: Target variable *T*, Data set *D*
Output: *CPC* of *T*

1 Set *CPC* as null
2 Repeat
3 <*F*, *assocF*>=MaxMinHeuristic<*T*, *CPC*>
4 Do Until *CPC* is not changed
5 If *assocF*≠0 then
CPC=*CPC*∪*F*
6 End do
7 For all *x*∈*CPC* do
8 if *S*∈*CPC.s.t.Ind(X;T|S)* then
CPC=*CPC*\{*X*}
9 End for
10 Return *CPC*

Subprogram MaxMinHeristic*(T, CPC)*

Input: Target variable *T*, subset *CPC*
Output: The max variable with minimum correlation with *T* in *CPC*

1 *assocF* = max*x*∈*vminAssoc(X; Y|CPC)*
2 *F* = arg_max*x*∈*vminAssoc(X; Y|CPC)*
3 Return *F*, *assocF*

However, there exists a problem in this algorithm. When removing the undesired variables in the second step, the candidate set will become inaccurate for the next searching cycle because of the different values of independent statistical information. In order to acquire the accurate network structure, the usual approach is to set the confidence value of independent testing as 1, thus reducing the numbers of undesired variables but also missing some search space.

B. Integrating PSO and GA

PSO simulates the process of searching food by a group of birds. Each bird is seen as a particle, which is probably the feasible solution of the problem. The birds always change their position and speed when flying in the sky. As discovered, during the searching process, the birds are scattered at first but gradually they gather together in line. This line is from side to side, up and down in the sky. Finally they achieve the food.

In PSO, all solutions in the optimization problem are seen as 'particle'. Each particle is corresponding to an fitness value decided by the optimization function. There is a speed value for each particle, determining the moving direction and distance. Every iterative process is to trace two extreme values: the local optimal value *pbest* from the individual particle and the global optimal value *gbest* within the whole groups. Until these two values are found out, the ith particle is updated according to below equations[3].

$$v_{ij}(t+1)=w*v_{ij}(t)+c_1r_1(t)(p_{ij}(t)-x_{ij}(t))+c_2r_2(t)(p_{gj}(t)-x_{ij}(t))$$
$$x_{ij}(t+1)=x_{ij}(t)+v_{ij}(t+1) \quad (3)$$

In the above equations, the subscript *j* represents the *j*th dimension of the particle. *i* represents the *i*th particle. The generation number is denoted by *t*. The constants *c1* and *c2* represent the moving step size, range in (0,2). r_1~*U(0,1)*, r_2~*U(0,1)* are two independent random functions. For updating particles by these equations, *c1* is to adjust the step length towards the best position of the particle itself, while *c2* is to adjust the step length towards the global best position within the group[4].

PSO is a kind of evolutionary algorithm. Similar to GA, starting from random solution, it is to find out the best solution through iterations. In essence, its approach is looking after the global best solution by tracing the current best value. PSO has no operation of crossover and mutation. Due to its simple calculation and rapid convergency, it is effective for a lot of global optimization problems. But PSO is not suitable for discrete optimization problems, as it is easy to result in local optimum. Instead, GA is to search the optimal solution by simulating the process of natural evolutionary. It can automatically direct to the optimized search space by probabilistic optimization method. But GA always results in premature convergency.

In order to improve the problem of randomly updating particles in PSO, we can integrate PSO and GA in Equation (3)[5]. As for the *w* in the first item of the equation, it reflects the global searching capacity of the particle. The value of *w* decides whether the particle can jump out from the local minimum point, which essentially the same with the mutation process in GA. The second and the third items are corresponding to the exchange of public and local particles respectively, similar to the crossover operation of GA. Therefore, it can be considered that the mutation operation replaces the first item, and the crossover operation replaces the second

and third items. Thus the particles can be updated more effectively. By integrating these two algorithms, it not only owns the advantage of simple calculation and global optimization of PSO, but also acquires the capacity of parallel computation from GA. This hybrid algorithm can avoid randomness for updating particles and improve the searching efficiency and accuracy.

a) Coding and Crossover

Based on the framework of the network structure worked out by MMPC, the edges set of the undirected graph is initially prepared. Now we transform the graph *S* to a matrix *C* as shown in Fig. 1. If there is an edge connecting node *i* and *j*, then C_{ij}=1; otherwise C_{ij}=0[6].

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

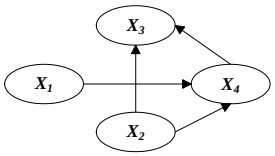


Figure 1. Network Structure

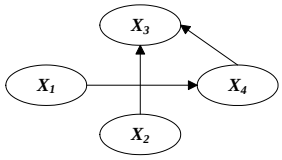
As shown in Fig. 1, the coding of the Bayesian network is 0001001100000010, corresponding to matrix *C*. The coding method is to change the matrix to row vectors, using two-point crossover as Fig. 2.

Particle A：0110↑0001↑00010000 → 0110001100010000
Particle B：0001↑0011↑00000010 → 0001000100000010

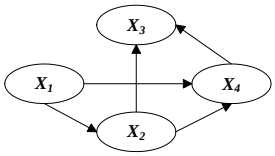
Figure 2. Two-point crossover

b) Mutation

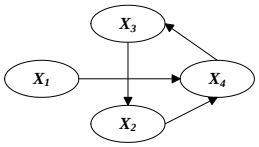
The mutation method in our proposed algorithm includes three types of operation: add edge, delete edge, and reverse edge. These 3 types of mutation operation are shown in Fig. 3.



(1) delete edge



(2) add edge



(3) reverse edge

Figure 3. Mutation Operation

c) Fitness Function

As for a random node x_i, there exists its parent node *Pa(i)*. The fitness value is calculated by the fitness function as Equation (5)[7]. The meaning of q_i, r_i, N_{ij}, N_{ijk} in this equation is the same as equation (2).

$$P(i, Pa(i)) = \prod_{j=1}^{qi} \frac{(r-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{ri} N_{ijk}! \qquad (5)$$

Each probable network structure is seen as a 'particle' in our proposed hybrid algorithm. We can calculate the fitness value of the particle using Equation (5). If the end condition of the algorithm is not fulfilled, the operation of crossover and mutation mentioned above will be repeated. A new generation of particle will be worked out and loop back to be tested by the end condition again.

d) Algorithm Description

Based on the network framework generated by MMPC, the initial model of the network graph is produced. By the operation of crossover and mutation, such as add edge, delete edge and reserve edge on the initial graph, the similar matrices from CPC are created as the initial group of particles. Each probable network structure is seen as a 'particle' in PSO. The crossover and mutation operation will repeat in the iteration until meeting the end condition of the algorithm. The hybrid optimization algorithm is detailed as below:

Hybrid optimization algorithm

Input: *CPC*
Output: Bayesian network structure *S*

1 Set *S* as null
2 Initialize particles *Par[n]* basing similar matrices from *CPC*
3 Call ***PSO***
4 While (*T* < T_{max} (maximum iteration number))
5 For 1 to *n*
6 Call ***GA***
7 End
8 Call ***PSO***
9 If new *Gbest* from step8 > old *Gbest* from step3 then
Retain new *Gbest* and *ParG*
10 End
11 Transform *ParG* to the output network structure *S*

Subprogram PSO

Input: Initial particles *Par[n]*
Output: global best fitness value *Gbest*,
global best particle *ParG*

1 For 1 to *n*
2 *Par.Fit* = CalculateFit(*Par[n]*) (fitness value)
3 If *Par.Fit* > *Pbest* (local best fitness value) then
ParP[n]= Par[n] (new local best particle)
Pbest[n] = *Par.Fit* (new local best fitness value)
4 End
5 *Gbest* = Max*(Pbest[n])* (global best fitness value)
6 Choose global best particle *ParG* = (*Par[n]* with *Gbest*)

Subprogram GA

Input: initial particles *Par[n]*,
local best particles *ParP[n]*,
global best particles *ParG*

Output: newly updated *Par[n]*

1 Encoding *Par[n]* and *ParG[n]*
2 Two-point crossover between *Par[n]* and *ParG*
3 Two-point crossover between *Par[n]* and *ParP[n]*
4 Transform new *Par[n]* to matrix for mutation
5 Output new *Par[n]*

IV. EXPERIMENTS

The data sample is the totally 11809 customers' personal records selected from the database of a national bank. Every customer is described by below variables and status:

- Sex: Male, Female
- Age: 18-30, 31-40, 41-50, 51 and above
- Economic Level: Low, Mid, High
- Loan Record: Yes, No
- Desire of purchasing funds: Yes, No

The target is to find out the factors affecting the desire of purchasing funds.

Table I summarizes the numbers of data sample grouped by every different value of the 5 variables. For example, the first number represents 17 people within the group (x1=Male, x2=18-30, x3=Low, x4=Yes, x5=Yes). The second number represents 336 people within the group (x1=Male, x2=18-30, x3=Low, x4=Yes, x5=No). Accordingly, following data in the table is given based on the values in left-to-right sequence of each variable.

In this experiment, the crossover rate is set as 0.7 and the mutation rate is set as 0.1. The number of particles is initialized as 20. The end condition for stopping searching is that the fitness value is not changed in the latest 100 continuous iterations. The max iteration number is 2000. We record the evolution results every 10 generations. The results will be compared to the general GA and the general PSO. Considering the randomness of initial particles, we take the average of 10 times results for each algorithm.

The experimental results show that the hybrid optimization algorithm is more effective than the other two. It can work out the best network structure until 470 generations, while this record of GA is 1380 and PSO is 1160. As shown in Fig. 4, the general GA has once worked out the best result at about 600 generation, however it is not the real best result. It is due to premature convergence. In contrast, the hybrid optimization algorithm can avoid this problem successfully.

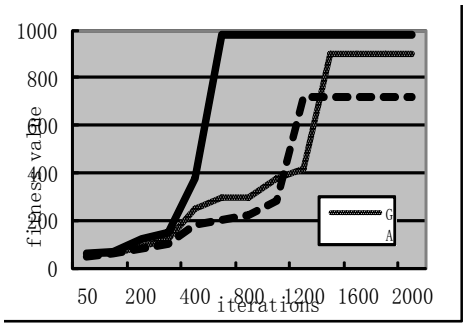
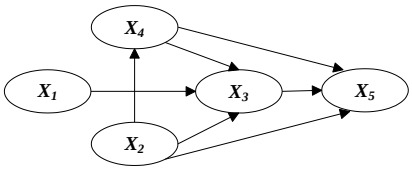


Figure 4. Comparison of efficiency

The corresponding Bayesian network structure resulted from the proposed algorithm is shown in Fig. 5. The result shows that economic level is affected by sex, age and loan record. Finally, the desire of purchasing funds is affected by age, economic level and loan record.



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

Figure 5. Network structure

The above network structure is the same as the target customer model for fund sales of the bank. This model can work out the target customers with high desire of purchasing funds, in order to raise the success rate of sales. The comparison of the actual sales record of the bank and the result from the model is shown in Table II. Table III shows that the model has the correct rate more than 97%.

TABLE I.
NUMBER OF PEOPLE GROUPED BY ALL STATUS

17	362	26	77	22	220	46	85	25	139	51	67	23	80	62	56
15	219	40	97	20	214	77	108	25	128	106	105	30	92	132	72
21	153	60	104	19	133	87	123	30	105	161	113	19	55	211	86
17	35	52	70	18	60	145	103	22	54	237	78	21	30	427	67
18	441	22	57	18	325	27	60	21	229	33	48	26	109	41	37
24	272	42	74	32	249	60	101	25	177	75	98	28	126	85	63
20	150	49	85	26	206	88	103	25	187	104	113	33	94	155	90
19	37	49	71	18	83	123	89	25	61	243	94	26	62	373	111

TABLE II.
COMPARISON OF ACTUAL RECORD AND EXPERIMENT RESULT

Purchasing Desire	Actual record	Experimental result
Yes	3054	3202
No	8755	8607
Total	11809	11809

TABLE III.
STATISTIC OF EXPERIMENTAL RESULTS

Purchasing Desire	Correct records	Correct rate	Error records	Error rate
Yes	2997	93.60%	205	6.40%
No	8550	99.34%	57	0.66%
Total	11547	97.78%	262	2.22%

V. CONCLUSION

The paper presents a hybrid optimization algorithm integrating MMPC, GA and PSO for Bayesian network structure learning. Using the network framework constructed by MMPC, the proposed algorithm applies GA and PSO in score-search. With advantage of GA and PSO, the initial particles can be selected effectively. Experimental results show that the proposed algorithm offers favorable learning capacity and convergence rate. The proposed algorithm provides a new way in practical application of Bayesian network. The algorithm needs professional experience to set up initialized crossover rate and mutation rate. The future work is to find a way that can set up initial parameters by machine learning.

ACKNOWLEDGMENT

The work was supported by National Natural Science Foundation of China (No. 61106019), Funds from Science and Information Technology Bureau of Guangzhou (No. [2013]163) and Funds from Science and Technology of Guangdong Province (No. 2012B091000172).

REFERENCES

[1] COOPER G, HERSKOVITSE. A Bayesian method for the induction of probabilistic networks from data [J]. machine learning, 1992,(9). p. 309-347.
[2] Tsamardinos I, Brown L F, Aliferis C F. The max-min hill-climbing Bayesian network structure learning algorithm [J]. Machine Learning, 2006, 65,(1). p. 31-78.
[3] Gao Shang, Yang Jingyu. Swarm intelligence algorithms and applications [M]. Beijing: China WaterPower Press, 2006. p. 6-10.
[4] Liang Jun. Research on particle swarm optimization applied in optimization problems [J]. Master degree theses of Guangxi normal university, 2008.
[5] Xu Lijia. Hybrid optimization for Bayesian network structure learning. Academic journal of Zhejiang university, 2009, (5).
[6] Junying Meng, Jiaomin Liu, Juan Wang, Ming Han. Target Tracking Based on Optimized Particle Filter Algorithm[J]. Journal of Software. 2013, Vol 8, No 5.
[7] Ming Li, Bo Pang, Yongfeng He, Fuzhong Nian. Particle Filter Improved by Genetic Algorithm and Particle Swarm Optimization Algorithm[J]. Journal of Software. 2012, Vol 7, No 6.
[8] Zhao Xuewu. Bayesian Network structure learning based on topological order and quantum genetic algorithm [J]. Application of Computer. 2013, (6)
[9] Tao Dong, Wenqian Shang, Haibin Zhu. An Improved Algorithm of Bayesian Text Categorization[J]. Journal of Software. 2011, Vol 6, No 9.

Junyi Li (1982-, Guangzhou, China), received her B.S degree in information management from South China Normal University, MSc degree in computer technology from Guangdong University of Technology, China.

She worked as a senior software engineer for HSBC and as a data analyst for China Guangfa Bank. Currently she is a lecturer in computer engineering department of Dongguan Polytechnic, China.

Ms. Li also acquired the qualification of senior information system management of China. She is a member of Guangdong Computer Society. Her interest includes data mining, artificial intelligence and algorithms design.