

Applying MDL in PSO for Learning Bayesian Networks

Shu-Ching Kuo

Department of Leisure and
Information Management,
Taiwan Shoufu University,
Tainan, Taiwan

Hung-Jen Wang

Department of Computer
Science and Multimedia Design,
Taiwan Shoufu University,
Tainan, Taiwan

Hsiao-Yi Wei

Institute of Information
Management, National Cheng
Kung University,
Tainan, Taiwan

Chih-Chuan Chen

Department of Leisure and
Information Management,
Taiwan Shoufu University,
Tainan, Taiwan.
Department of Industrial and
Information Management,
National Cheng Kung
University,
Tainan, Taiwan

Sheng-Tun Li*,
Member, IEEE

Institute of Information
Management, National Cheng
Kung University, Tainan,
Taiwan
Department of Industrial and
Information Management,
National Cheng Kung
University,
Tainan, Taiwan

Abstract—Since learning Bayesian networks from data is difficult, a new approach is proposed. The particle swarm optimization (PSO) and minimum description length (MDL) are combined to obtain a suitable Bayesian network. MDL is the fitness function in this learning algorithm to evaluate the goodness of the network. By adopting MDL, the balance between simplicity and accuracy is assured, which enables the optimal solution for complex models to be found in reasonable time. Based on the MDL principle, the PSO is used to enhance the structure learning in Bayesian networks. Moreover, conditional probabilities associated with the Bayesian networks are then statistically derived from these data. In the end, the Stroke data set is used for testing the efficiency and effectiveness of the stable network. Experimental results show that the proposed approach has a good accuracy than the comparative methods.

Keywords— Bayesian networks; particle swarm optimization; minimum description length

I. INTRODUCTION

Bayesian networks provide an efficient method for representing conditional probabilities among a set of variables. A Bayesian network is a graphic representation that encodes probabilistic relationships among variables. Since it only depicts the known conditional dependencies among variables rather than consider all variables in the domain, it provides efficiency for acquiring and representing domain knowledge [1]. Because of the ability to properly express expert knowledge and to handle well for the uncertain-reasoning problems, Bayesian networks have increasingly gained popularity in the community of artificial intelligence in recent years [2].

Many approaches have been proposed to induce the structure of Bayesian Networks. Typically, a Bayesian network is constructed by eliciting knowledge from domain experts. It is however a labor intensive process and may generate some biases. To reduce these biases due to subjective judgments of experts and increase efficiency, learning a Bayesian network from collected data or past observations has

become an active area of research. Several heuristic searching techniques have been used to create a Bayesian network. Among those heuristics, evolution computation has been intensively researched and proved to be effective in learning Bayesian networks [3-4].

Numerous evolution techniques such as ant colony optimization, genetic algorithm and evolutionary programming were developed for Bayesian network structure discovery from data. Compared with these algorithms, particle swarm optimization (PSO) has strong sociological activities that promote information sharing and, in addition, it stores the previously best explored areas of the search space among all particles. By well communicating with others, each particle thus shares the information and has more opportunities to trend to better places. Moreover, PSO converges rapidly and requires only primitive mathematical operators. It is easy to implement but has great capability to handle optimization problems with multiple local optima reasonably well with fairly high accuracy. PSO has been found being robust in solving nonlinear and high dimensional problems [5-6].

In this paper, PSO and minimum description length (MDL) are combined to obtain a well-defined Bayesian network to assist domain experts in making decisions. Since the experts may be uncertain about their diagnoses, the results predicted by Bayesian networks will be a direction to help them consider how to solve the uncertainty. Even if the experts can make their own decisions, a Bayesian network can be taken as a reference to make them assure the results and even inspire them to think about some implications behind the relations among random variables. Moreover, PSO algorithm is used to learn the structure of a Bayesian network and the MDL is applied as the fitness function of the algorithm so that the final network will be useful and having a good accuracy at the same time. Further, the less complicated structure allows the domain experts to easily understand and quickly catch the causes and effects between the variables.

The remainder structure of the research is organized as follows: Section II briefly addresses the concept of Bayesian networks, PSO algorithms and MDL metric. It also discusses how they work together in various different domains. The proposed scheme is shown in Section III. The experimental

*Corresponding author: stli@mail.ncku.edu.tw

results are shown in Section IV. Finally, some conclusions are described in Section V.

II. RELATED WORKS

A. Particle Swarm Optimization

Particle swarm optimization (PSO) is a searching algorithm proposed by Kennedy and Eberhart [7]. Social behaviors of bird flocking motivated them to search and observe the effect of collaboration of species onto achieving their goals as a group. In PSO, a number of candidates, called particles, are placed in a D -dimensional search space looking for the optimal position to land. The particles will memorize the locations where they have their best locations and make this information available to its neighbors. Each particle adjusts its position by combining its own experience and the experience of other swarms as time passing. The next iteration takes place after all particles have been moved. In the end, the swarm is able to move close to an optimum of the fitness function.

Figure 1 shows the flow diagram of original PSO algorithm, and the corresponding Algorithm is given in Algorithm 1 [8]. In PSO, three parameters determine a particle's position including the current position $\bar{x}_i$, the previous best position $\bar{p}_i$, and the velocity $\bar{v}_i$. Each particle keeps track for the best solution called pbest. On the other hand, gbest is the best value tracked by the global version of the entire swarm. $U(0, \varphi_i)$ represents a vector of random numbers uniformly drawn on the interval (0, 1).

Algorithm 1. Original PSO algorithm.

- Step 1. Initialize a population array of particles with random positions and velocities on D dimensions in the search space.
- Step 2. Loop
- Step 3. For each particle, evaluate the desired optimization fitness function in D variables.
- Step 4. Compare particle i -th fitness evaluation with its pbest _{i} . If current value is better than pbest _{i} , then set pbest _{i} equal to the current value, and $\bar{p}_i$ equal to the current location $\bar{x}_i$ in D -dimensional space.
- Step 5. Compare particle i -th fitness evaluation with the population's overall previous best gbest. If current value is better than gbest, then reset gbest to the current particle's array index and value.
- Step 6. Change the velocity and position of the particle according to the following equation:
$$\bar{v}_i \leftarrow \bar{v}_i + U(0, \varphi_1) \otimes (\bar{p}_i - \bar{x}_i) + U(0, \varphi_2) \otimes (\bar{p}_g - \bar{x}_i) \quad (1)$$

$$\bar{x}_i \leftarrow \bar{x}_i + \bar{v}_i \quad (2)$$
- Step 7. If a criterion is met (usually a sufficiently good fitness or a maximum number of iterations), exit loop.
- Step 8. End loop

The parameters φ_1 and φ_2 are often called acceleration coefficients which determine the weight of the random forces

in the direction of personal best $\bar{p}_i$ and global best $\bar{p}_g$. φ_1 represents the attraction that a particle has toward its own success while φ_2 represents the attraction that a particle has toward the success of its neighbors. It is almost ubiquitously adopted to set $\varphi_1 = \varphi_2 = 2.0$. Each component of $\bar{v}_i$ is kept within the range $[-V_{\max}, +V_{\max}]$. The choice of the parameter $V_{\max}$ requires consideration since it appears to influence the balance between exploration and exploitation. If $V_{\max}$ is too high, particles may fly past the good positions. On the other hand, the particles could become trapped in local optima if $V_{\max}$ is too low.

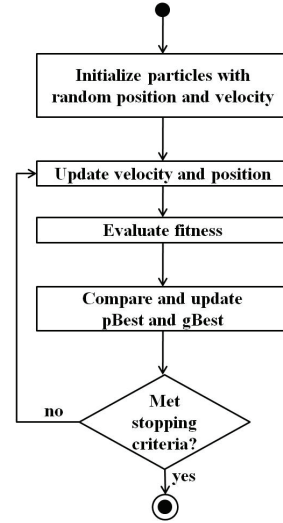


Figure 1. Particle Swarm Optimization flow diagram

B. Minimum Description Length

The minimum description length (MDL) principle proposed by Rissanen [9] is a learning criterion that selects the rule to find the best balance between the simplicity of a model and the goodness of fit to examples based on a data compression context. The fundamental insight behind the MDL Principle is that any regularity in a given set of data can be used to compress the data, that is, to describe it using fewer symbols than the number of symbols needed to describe the data literally [10]. Therefore, it embodies a form of Occam's Razor, a principle that claims the simpler one is the better when two competing theories that make exactly the same predictions. This idea can be applied to all sorts of inductive inference problems, and it turns out to be most fruitful in problems of model selection and, more generally, dealing with overfitting [11].

C. Bayesian networks

The Bayesian network which can also be referred to as a probabilistic network, belief network, Bayesian Belief Network and causal network was introduced for representing and reasoning with uncertainty [12]. In general, it consists of a graphical structure G which is a directed acyclic graph (DAG) and an associated probability distribution Pr . There are two parts to the Bayesian network. One is the qualitative relationships between the random domain variables, and the

other is the quantitative part, the probabilities over the variables.

Figure 2 shows a simple Bayesian network. Formally, let $N = [N_1, \dots, N_n]$ be a set of nodes. Each node which is represented in the graph corresponds to a discrete random variable in the domain. The variables in a Bayesian network could be everything such as people, events and may be continuous or discrete. Each node has a number of states, that is, a set of probable values for each variable. There is a parent set, denoted by Π_p , for each node. For example, according to Figure 2, lung cancer may be caused by smoking or air pollution. Therefore, smoking and pollution together form the parent set of the node cancer.

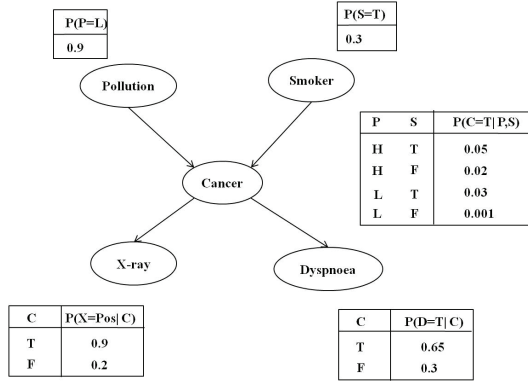


Figure 2. A simple Bayesian network representing the lung cancer problem

If a node has no parent nodes, its parent set would be empty and its probability table gives the marginal probabilities of the node. The directed arrows correspond to the dependence relationships between variable-pairs. It is usually called edge or arc. A node is taken to be dependent of its parents and children in the digraph, but is conditionally independent of any of its non-descendants given its parent.

In the Bayesian network, prior probabilities are assigned to source nodes (i.e., nodes without any incoming arcs), and conditional probabilities are associated with arcs. If a node X has only one parent Y , node X 's probability table will contain the conditional probability $P(X|Y)$. Moreover, if X has multiple parents $(Y_1, Y_2, \dots, Y_k)$, the table will contain the conditional probability $P(X|Y_1, Y_2, \dots, Y_k)$.

This quantitative part encodes the joint probability distribution of these variables. The joint probability of any instantiation of all variables can be calculated as follows:

$$P(N_1, N_n) = \prod_i P(N_i | \text{Parents}(N_i)) \quad (3)$$

where N_i represents the variables, and the value of the node is conditional only on the values of its parent nodes. Bayes' rule is used to update the conditional probabilities given evidence so the probabilities for all the nodes in the Bayesian network can be calculated. For example in Figure 2, the probability of the patient having cancer with smoking habit when the air

pollution is high and the result of X-ray is true is $0.005 - 0.9 / 0.005 - 0.9 + 0.95 - 0.2 = 0.191$.

Building a Bayesian network involves three tasks. The first task is to identify the domain variables to be captured with the values they may adopt. Once the variables are decided, the causality relationships between these variables are identified and analyzed, and are expressed in a graphical structure. After creating the qualitative architecture, the next important task is to obtain the probabilities for each variable. The three tasks are typically iterated in a cyclic fashion since building the Bayesian network is flexible and creative.

D. Applying the MDL principle to Bayesian networks

MDL is first adopted for evaluating Bayesian networks by Lam and Bacchus [13]. The MDL principle suggests that the rule g that minimizes the description length given examples should be the output of the learning algorithm. As used in Bayesian belief network, each rule is views as a possible network structure and sequence given the rule g can be seen as the value of the j -th attribute. So in Bayesian belief network, the MDL principle is based on the idea that the desired model of a collection of data items is the model that minimizes the sum of the length of the encoding of the model and the length of the encoding of the data given the model. Even if the minimal network can't be found, structures with lower total description lengths are to be preferred. Such structures are superior to structures with longer total description lengths, in the precise sense that they are either more accurate, or are less complex, or are closer in structure to the original network. Hence, the total description length provides a metric by which alternate candidate structures can be compared [14].

The minimum description length metric can be expressed as:

$$D_t(B) = \sum_{N_i \in N} D_t(N_i, \Pi_{N_i}) = \sum_{N_i \in N} (D_n(N_i, \Pi_{N_i}) + D_d(N_i, \Pi_{N_i})) \quad (4)$$

where $D_t(B)$ is the total length of a network which is based on two components: the network description length D_n and the data description length D_d . The network description length (D_n) includes:

1. A list of the parents of each node.
Suppose there are n nodes in the network, and each node has k different parents. For a node, $k \cdot \log_2(n)$ bits are needed to encode its parents.
2. The set of conditional probabilities associated with each node.

To encode this part, it firstly needs to count the total number of conditional probabilities of each node. The number of bits required to store the numerical value of each conditional probability is also required. The encoding length will be the product of the number of bits and the number of conditional probabilities. Suppose a node takes on s_i values, F_i different parents each of which with distinct s_j values and d represents the number of bits required to store a numerical value. Thus for a node, $d(s_i - 1) \prod_{j \in F_i} s_j$ bits are needed.

Due to above two mentioned parts, the total description length for a particular network will be defined as

$$\sum_{i=1}^n \left[k_i \log_2(n) + d(s_i - 1) \prod_{j \in F_i} s_j \right] \quad (5)$$

The data description length (D_d) is the length of the string encoding the database which can be expressed as

$$-N \sum_i p_i \log_2(q_i) \quad (6)$$

where N is the total number of records in the database, p_i is the true probability of an event and $-\log_2(q_i)$ is a codeword of length when run using this probability according to Huffman's algorithm. It is close to $-\log_2(p_i)$ since the probability q_i is determined by the network. Each event will have N_{p_i} occurrences in the database.

III. PROPOSED SCHEME

In this paper, construction of the Bayesian network is divided into two steps. The first step is network structure learning, and the second step is probability estimation. The whole process is represented in Figure 3. In the structure learning, a set of Bayesian networks are first randomly generated. Then, PSO is used to enhance these structures. After getting the best network, the probabilities associated with the networks are calculated according to the database. When the structure and probabilities are acquired, the final Bayesian network is obtained and can be used for prediction or classification.

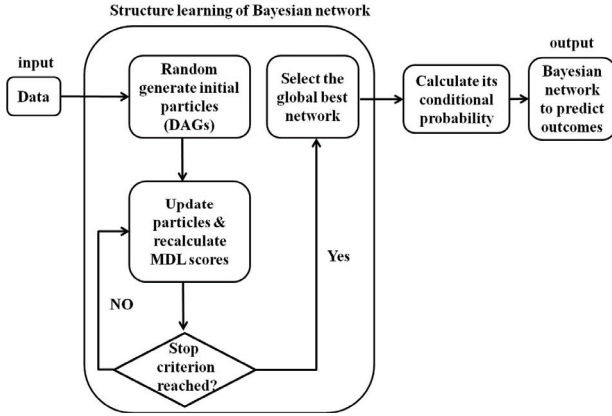


Figure 3. Bayesian Networks Learned by PSO

In structure learning, the PSO is adopted as a search algorithm which attempts to find an optimal Bayesian network. The search space of the networks is a DAG space, that is, every individual in the space is a directed acyclic graph (DAG). Before using PSO in a DAG space, several mathematical functions including (1) each particle's position, (2) each particle's velocity, (3) how to determine the new position of the particle, and (4) how to determine the new velocity of the particle should be defined.

In the research space, each particle represents a candidate Bayesian structure. The position of a particle is represented as an $n \times n$ adjacent matrix where n is the number of nodes in the problem. Elements in the matrix represent the dimension of the particle. The presence of zero in any element indicates that

these two nodes have no linkage and one indicates that there is a connecting edge between the two nodes. An example of a matrix and its corresponding Bayesian Network structure is shown in Figure 4.

Velocity is used to determine the new position of a particle. It can measure the difference between two positions and thus move the particle. The velocity of a particle is also defined as an $n \times n$ matrix. The elements in the matrix are represented as discrete values $\{-1, 0, 1\}$. A velocity with -1 in an element means that the existing edge between two specified nodes in the DAG will be removed in the next generation, 0 means that the edge between two nodes remains unchanged, and 1 means that a new edge will be added between two specified nodes. In the velocity for each element indicates whether the edge represented by the element is to be removed, added or left unchanged.

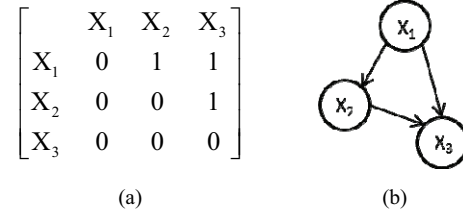


Figure 4. (a) Matrix of position. (b) Corresponding Bayesian network.

TABLE I. VELOCITY UPDATE

Previous position	Current velocity	Current position	Action
0	0	0	Unchanged
0	-1	0	Unchanged
0	1	1	Add
1	0	1	Unchanged
1	1	1	Unchanged
1	-1	0	Remove

Since the constriction factor has a better performance than inertia weight [15], it is used to compute the new velocity of a particle, which is defined as Eq.(7).

$$\bar{v}_i \leftarrow \chi (\bar{v}_i + c_1 * rand() * (\bar{p}_i - \bar{x}_i) + c_2 * rand() * (\bar{p}_g - \bar{x}_i)) \quad (7)$$

$$\chi = \frac{2}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}} \quad (8)$$

$$\varphi = c_1 + c_2 > 4 \quad (9)$$

In the initial generation, $\bar{v}_i$ is created randomly. Beginning from the second generation, it is updated by combining the current $\bar{v}_i$ and the cognitive and social influences respectively, working as the formula above. After updating the velocity, the value has to be translated into discrete values $\{-1, 0, 1\}$. To partition values of a continuous variable into a discrete variable, K equal lengths (equal intervals) is used since it is one of the most familiar methods to create partitions. K is chosen to provide a manageable number of discrete values, which gives a fair approximation of the continuous frequency distribution [16]. In this paper, K is set to 3.

In order to get the optimal network, a fitness function is necessary to evaluate the goodness of all candidate networks. In this paper, the Minimum description length (MDL) metric is applied to assess the models. In the MDL principle, the best model of a collection of data items is whose length is the minimum. The length is the sum of two parts: (1) the length of the encoding of the model, and (2) the length of the encoding of the data given the model. Both can be measured in bits. [13]

The structure learning of Bayesian network algorithm combines PSO and MDL, which includes two steps: create and update. The first step is the creation of a set of particles (DAGs) and initialization of the personal and global best positions. The second step is the calculation of velocities and update of the positions of the particles. The details of the algorithm are shown in Figure 5.

```

Let  $N$  be the set of all nodes.
Randomly generate  $K$  particles.
Set  $round$  to 0.
While  $round$  is smaller than the maximum number of
generations,
  Each particle is evaluated with the fitness function. The
  best position of each individual and the best position of
  the global swarm are determined by the MDL scores.
  For each particle,
    Determine the next velocity and position of each
    particle using the velocity update equation.
  For each node,
    Run the validation algorithm of the structure to
    check whether it violates the Bayesian network
    standards. If cycles are not generated by inserting
    new edges into the DAG, then add the edges. If it
    creates cycles, then delete the edges or skip the
    generation.
  Perform the fitness function to assess the goodness of
  the DAG.
  Increase  $round$  by 1.
Let the DAG with the lowest fitness value found in any
generation be the final best network.
Estimate the probabilities of the network.
Return this network as the result of the optimal Bayesian
network.

```

Figure 5. Algorithm for PSO to learn Bayesian networks

After the validations, each particle is again evaluated by the MDL metric. This can ensure that the best particle of this generation will be stored to the next generation. After running a preset number of generations, an optimal particle is acquired. Probabilities of the nodes in the network are then calculated as described in the next section. There are two types of probabilities in a Bayesian network, which include prior probabilities and conditional probabilities. If a node has no parents, it is assigned to the prior probability. The prior probability is measured as the number of cases of this event divided by the total number of cases in the database, which is shown as:

$$P(N_i = c) = \frac{N_{i,c}}{N} \quad (10)$$

If a node has parents, it is assigned to the conditional probability. The conditional probability can be calculated as follows:

$$P(N_i = c | \Pi_{N_i} = p) = \frac{N_{i,j,k} + 1}{N_{i,j} + r_i} \quad (11)$$

where r_i is the number of different values of variable N_i , $N_{i,j,k}$ is the number of cases in the database in which variable N_i has the value set c and Π_{N_i} has the value set p . $N_{i,j}$ is the sum of

cases of different value that the variable N_i takes on in the database, which can be instantiated as

$$N_{i,j} = \sum_{k=1}^{r_i} N_{i,j,k} \quad (12)$$

After acquiring all prior and conditional probabilities, the Bayesian network can be used for probabilistic inference which refers to the process of computing the posterior probability of a query variable after obtaining observable variables [17], which can be defined as

$$P(N_i | N_j) = \frac{P(N_i)P(N_j | N_i)}{P(N_{i1})P(N_j | N_{i1}) + P(N_{i2})P(N_j | N_{i2}) + \dots + P(N_{im})P(N_j | N_{im})} \quad (13)$$

where N_i is the query node, and N_j is the observable nodes. N_j is the child of N_i . $N_{i1}, \dots, N_{im}$ are states of node N_i .

IV. EXPERIMENTAL RESULTS

In order to show that the proposed method has a good performance, the Stroke data set which is collected from a neurological clinic is used. There are 350 instances gathered from 2009 to 2010, in which 176 instances are with stroke and 174 instances without stroke. The details of the data set are presented in Table II. Each instance comprises 11 attributes, including age, gender, obesity, hypertension, diabetes mellitus, coronary artery disease, left ventricular hypertrophy, atrial fibrillation, smoking, cholesterol and high-density lipoprotein. The class is whether the patient suffers from stroke. All attributes are binary except for age which is trinary.

MDL is a measure to evaluate the learned Bayesian network. The lower the measure, the better the network is. In the first generation, the MDL score of the model is around 4320.55. Then the MDL metric of Bayesian network declines as the PSO algorithm learns better network models. The optimal network appeared at the 1050th generation, with a MDL score of 4122.54. Due to the fast learning rate to achieve optimal model, the proposed method is efficient. The trend of MDL scores is presented in Fig. 6.

The algorithm runs with particles set to 20, 50, 100, 150, and 200, respectively. Executing the algorithm over different particles can evaluate whether a better scoring Bayesian network can be generated by more explorations. 5000 iterations are executed. Other parameters such as constriction factor are set to 0.729, and the two learning factors are set to 2.05. Different values of particles of PSO are conducted to study its effect on the stroke problem. Structural difference is the measure to evaluate Bayesian network structure. It is defined as: $\sum_{i=1}^n \phi_i$ where ϕ_i is the sum of the symmetric difference between the parent sets of node i in the learned network and the known network [3]. Ten trials of experiments are performed. The structural differences between the best Bayesian network structure found in each trial. There is no significant difference for various values of particles by using One-way analysis of variance (ANOVA) ($F = 1.464$, $P = 0.228$).

Accuracy of classification, sensitivity and specificity have been commonly used as indices to evaluate the quality of algorithms. Waikato Environment for Knowledge Analysis

(WEKA) system release no. 3.6.1 [18] thus is used as reference because it contains a large number of techniques divided into seven groups (Bayesian, based on functions, lazy, meta techniques, tree, rule and others). Support vector machine (SVM) and Bayes network are chosen to compare the classification performance with MDL-PSO. Parameters of each technique are the default values set in WEKA. The comparison results of Stroke breast cancer database are presented in Table III. The first row of Table III is the average accuracy of classification of each classifier. For the Stroke data set, MDL-PSO performs both better than SVM and BayesNet. The second and third rows are respectively sensitivity and specificity of the classifiers, which PSO-MDL got higher specificity than the other two classifiers. Sensitivity of PSO-MDL is lower than SVM but higher than BayesNet.

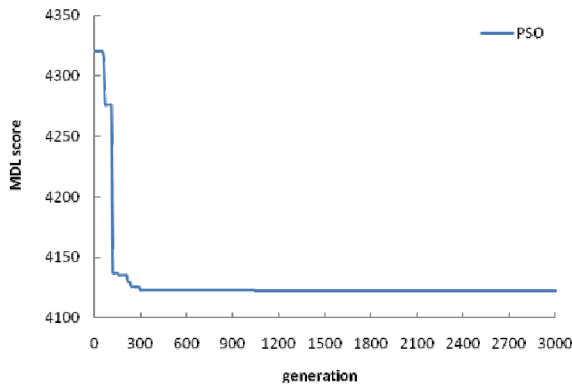


Figure 6. The MDL metric for Stroke data set

TABLE II. SUMMARY OF THE STROKE DATASET

Attribute	Data Type	Possible value
Age (<45, 45~65,>65)	Nominal	Young/Mid/Old
Gender	Nominal	Male/Female
Obesity (BMI>27)	Nominal	Yes/No
Hypertension	Nominal	Yes/No
Diabetes mellitus	Nominal	Yes/No
Coronary artery disease (>200)	Nominal	Yes/No
Left ventricular hypertrophy (<40)	Nominal	Yes/No
Atrial fibrillation	Nominal	Yes/No
Smoking	Nominal	Yes/No
Cholesterol(mg/dl)	Nominal	Normal/Abnormal
High-density lipoprotein(mg/dl)	Nominal	Normal/Abnormal
Class: Stroke or not	Nominal	Yes/No

V. CONCLUSION

PSO was adopted as the searching algorithm to learn the structure of Bayesian networks in this paper, and MDL was its fitness function. In addition, several constraints were used in order to keep the learning network from violating the standards of Bayesian networks. The learned Bayesian networks were also validated upon real data sets such as Stroke. The experimental results revealed that PSO-MDL was an efficient learning algorithm due to its fast learning rate. The accuracy is used to evaluate the effectiveness of PSO-MDL. As compared to SVM and BayesNet, the proposed method has a good accuracy.

TABLE III. THE COMPARISON RESULTS OF STROKE DATA SET

	MDL-PSO	SVM	BayesNet
Average accuracy	73.81%	68.86%	67.14 %
Sensitivity	70.5%	73.3%	65.3%
Specificity	65.5%	64.4%	63.2%

REFERENCES

- [1] G. F. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," *Artificial Intelligence*, vol. 42, pp. 393-405, 1992.
- [2] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Francisco: Morgan Kaufmann, 1988.
- [3] M. L. Wong, and K. S. Leung, "An Efficient Data Mining method for learning Bayesian networks using an evolutionary algorithm-based hybrid approach," *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 378-404, 2004.
- [4] X. Li, S. Yuan, and X. He, "Learning Bayesian Networks Structures Based on Extending Evolutionary Programming," *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, pp. 1594-1598, 2004.
- [5] A. Salman, I. Ahmad, and S. Al-Madani, "Particle swarm optimization for task assignment problem," *Microprocessors and Microsystems*, vol. 26 (8), pp. 363-371, 2002.
- [6] T. Du, S. S. Zhang, and Z. Wang, "Efficient Learning Bayesian Networks Using PSO," *International conference on Computational intelligence and security*, vol. 3801, pp. 151-156, 2005.
- [7] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995.
- [8] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization: An overview," *Swarm Intelligence*, vol. 1, 2007.
- [9] J. Rissanen, "Modelling by the shortest data description," *Automatica*, vol. 14, pp. 465-471, 1978.
- [10] J. Suzuki, "Learning Bayesian Belief Networks Based on the MDL Principle: An Efficient Algorithm Using the Branch and Bound Technique," *IEICE Trans. Inf. Sys.*, vol. E82-D (2), pp.3 56-367, 1999.
- [11] P. Graunwald, "A tutorial introduction to the minimum description length principle," In P. Graunwald, I. J. Myung, and M. Pitt, editors, *Advances in Minimum Description Length: Theory and Applications*. MIT press, 2005.
- [12] M. J. Druzdel and L. C. van der Gaag, "Building Probabilistic Networks: "Where Do the Numbers Come From?," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12(4), pp. 481-486, 2000.
- [13] W. Lam and F. Bacchus, "Learning Bayesian Belief Networks: An Approach Based On The MDL Principle," *Computational Intelligence*, vol. 10, 1994.
- [14] W. Lam, "Bayesian Network Refinement Via Machine Learning Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20(3), pp. 240 - 251, 1998.
- [15] R. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," *Proceedings of the 2000 congress on Evolutionary Computation*, pp. 84-88, 2000.
- [16] E.J. Clarke and B.A. Barton, "Entropy and MDL Discretization of Continuous Variables for Bayesian Belief Networks," *International Journal of Intelligent Systems*, vol. 15, pp. 61-92, 2000.
- [17] N. L. Zhang and D. Poole, "Exploiting causal independence in bayesian network inference," *Artificial Intelligence*, vol. 5, pp. 301-328, 1996.
- [18] I. H. Witten and E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques," San Francisco: Morgan Kaufmann, 2000.