Learning Bayesian Network Classifier Based on Artificial Fish Swarm Algorithm

Chun-Feng Wang, and Kui Liu

***Abstract*—Classification is an important task in data mining, which has been successfully applied to many areas. Bayesain network classifier aims to compute the class with the highest probability given a case. Since learning Bayesian network classifier from a dataset can be viewed as an optimization problem, heuristic algorithms may be used to find high-quality networks in medium or large scale problems. In this paper, we present a new artificial fish swarm algorithm for learning Bayesian network classifier. In this algorithm, an unconstrained optimization problem is established firstly. Its optimal solution is an undirected graph, which can be used to reduce the search space. Then, three behaviors of the artificial fish swarm are defined. Finally, the detailed description of the algorithm is given. In the experimental of the paper, the performance of the proposed algorithm is compared with other three classifiers. The results show that the proposed algorithm is effective.**

***Index Terms*—Bayesian network; structure learning; classification.**

I. Introduction

DATA mining is an active research area involving the development for exacting interesting knowledge from real-world datasets. Classification is a central problem in data mining field, where the goal is to build a classifier to predict. Classification includes supervised classification and unsupervised classification. Generally speaking, the aim of supervised classification is to assign labels or categories to instances described by a set of features. The classifier presented by this paper is a supervised classifier.

Since a Bsyesian network(BN)[1-3] can provide a graphical model for encoding relationships, such as dependencies and conditional independencies among variables, and for inferring probabilistically about variables, it is very suitable for classification. Bayesian network classifiers have many advantages over other classification methods. (1) They can offer an explicit, graphical and interpretable representation of uncertain knowledge based on the sound concept of conditional independence. (2) Since they output a probabilistic model, they can provide a confidence measure on the chosen predicted label. (3) Due to the model expressiveness of Bayesian network classifiers, they can easily adapt to feature selection methods and handle missing data in learning and inference phases. Meanwhile, they can deal with more complex classification problems in any type of domain (discrete, continuous, and mixed data), with undetermined labels, partial labels, many class variables to be simultaneously predicted, etc. (4) These algorithms(includes Naïve-Bayes Bayes and tree-augmented Naïve-Bayes Bayes,etc.)are easily implemented.

In these classifiers, Naïve-Bayes is the simplest type of Bayesian classifier[4]. Its performance can be comparable with other classification methods, such as decision trees and neural network. Nonetheless, since the attribute independency assumption is not realistic in many datasets, many attempts have been proposed to improve the performance of Naïve-Bayes by extending it to more sophisticated types of probabilistic graphical models. For example, to enrich the network structure, a tree augmented naïve Bayes (TAN) was developed[5]. According to this approach, a tree structure is applied for the classification to be achieved. The tree structure has advantage of low degree of complexity, along with the ability to avoid over fitting problems. However, it restricts the number of parents, other than the classification node, to exactly one single parent for each node, which turns out to be a strong constraint. So, the resulting structure appears to neglect the case where a variable is correlated with several other variables. Cheng et al. presented Bayesian network augmented naïve Bayes(BAN)[6], which further expanded the dependency relationship between any two attribute nodes. Madden proposed a general Bayesian network Bayes(GBN), which is an unrestricted Bayesian network[7]. In GBN, all nodes may have a parent and also be a child node of some attribute nodes. GBN is well reflected in the actual data.

As we all know, it is NP-hard to learn the optimal Bayesian network (BN) structure from a dataset, and the search methods used usually are greedy and deterministic, which prone to get stuck in local optima. So, many stochastic heuristic search algorithms have been proposed on medium and large size problems recently. Such stochastic algorithms perform a more global search that is less likely to get stuck into local optima to build high-quality Bayesian network classifiers in an acceptable computational time. For example, by using the particle swarm optimization algorithm, three different particle swarm data mining algorithms were presented and tested against a genetic algorithm and a tree introduction algorithm(J48)[8]. A hybrid adaptive particle swarm optimization for learning Bayesian classifier was proposed, which can exhibit an improved capability to eliminate spurious features from large datasets and aid researchers in identifying those features that are responsible for achieving higher classification accuracy[9]. A framework was presented to quantify which classifier is most effective at mining a given dataset in term of accuracy[10]. A new method for data mining particulary in classification tasks is developed based on ABC algorithm[11]. A new approach was pro-

Manuscript received March 8, 2015; revised June 28, 2015. This paper is supported by the National Natural Science Foundation of China (U1404105), the Key Scientific and Technological Project of Henan Province(142102210058); the Youth Science Foundation of Henan Normal University(2013qk02); Henan Normal University National Research Project to Cultivate the Funded Projects (01016400105).

Chun-Feng Wang is with College of Mathematics and Information Science, Henan Normal University,Xinxiang, 453007, PR China; Henan Engineering Laboratory for Big Data Statistical Analysis and Optimal Control, School of Mathematics and Information Sciences, Henan Normal University e-mail:wangchunfeng10@126.com.

Kui Liu is with College of Mathematics and Information Science, Henan Normal University, Xinxiang, 453007, PR China.

posed, which can be jointly used with the $K2$ algorithm pertinent to the structure learning of Bayesian classifiers[12]. A novel Bayesian classification algorithm (ABC-Miner) was proposed, which learns the structure of a BAN with at most k-dependencies from dataset using ACO technique for optimization[13].

This paper presents a new algorithm for learning Bayesian network classifier based on artificial fish swarm algorithm. In this algorithm, for reducing the searching space, the strategy of [14] is adopted firstly. Then, some rules of artificial fish swarm algorithm for learning Bayesian network are introduced. Based on the above, the method proposed in this paper is given.

The rest of the paper is organized as follows. we give an overview on the basic concepts of Bayesian Networks, then we describe various BN classifiers. In Section 3, our algorithm is described. In Section 4, the details of the experimental procedures and results of the proposed algorithm are given.

II. Bayesian networks

A. Overview on Bayesian networks

A Bayesian network $B =< G, \theta >$ is a directed acyclic graph G, which is a powerful tool for knowledge representation and inference that encode variables dependence and independence relationships. More precisely, Bayesian network (BN) represents a model of the joint probability distribution of n random variables $V = \{v_1, v_2, \cdots, v_n\}$, and a set of conditional probability tables (CPTs), one for each variable, is computed to represent the parameters θ of the network. With the independence statements encoded in the DAG, the joint probability function can be computed as the following formula:

$$p(v_1, v_2, \cdots, v_n) = \prod_{i=1}^{n} p(v_i | Pa(v_i), \theta, G), \qquad (1)$$

where G is the DAG that represents the structure of the BN, and $Pa(v_i)$ are the parents of variable v_i in G.

The process of learning (or constructing) a Bayesian network from a dataset D contains two phases, namely learning the network structure, and then learning the parameters of the network. Compared to the former, the latter phase is a relatively easy phase, so, we mainly consider how to find the best network structure. The purpose of the network structure-learning phase is to find G that maximizes $P(D \mid G)$ for a given D. A popular approach to that phase consists of using a scoring function, f, that evaluates each candidate G with respect to D, searching for the best network structure according to that score, i.e. find a G^* such that

$$G^* = \arg \max_{G \in G_n} f(G : D), \qquad (2)$$

where $f(G : D)$ is the scoring function measuring the degree of fitness of any candidate DAG to the data set, and G_n is the family of all the DAGs defined on V. A desirable and important property of a scoring metric is its decomposability in the presence of full data. Various scoring metrics (such as BIC, K2 and AIC) for learning a BN structure have been proposed in the literature [15-17]. In this paper, the Bayesian information criterion (BIC) score is used to benchmark the mete-heuristic optimization strategies.

During the past tears, many algorithms have been presented. In these algorithms, a well-known greedy and deterministic algorithm for learning a BN structure is Algorithm B [18]. This algorithm is initialized with an empty DAG (i.e., an edge-less graph structure) and iteratively adds, to the current network structure, the edge that leads to the maximum increase in the scoring function f, subject to the constraint that no directed cycles are included in the graph. The algorithm stops when there are no more valid edges to be added, or when adding any valid edge does not increase the value of the scoring function.

For further information about Bayesian networks, the reader may refer to [19], which provides an excellent overview of the subject.

B. Bayesian networks classifiers

BN classifiers are a special kind of BNs where the class attribute is treated as a special variable in the network. The aim of BN classifier is to compute the posterior probability of each value c in the class variable C given a case v (an instance of the input attributes V) using network G, then label this case with the class value having the highest posterior probability. This process can be completed by the following formula:

$$C(v) = \text{argmax}_{c \in C} P(C = c \mid V = v, G) \qquad (3)$$

$$\overbrace{P(c \mid V = v, G)}^{perior\ probability} \propto \overbrace{P(v \mid C = c, G)}^{likelihood} \times \overbrace{P(C = c)}^{prior\ probability}$$

where $\propto$ is the mathematical symbol for the proportionality relationship. Different types of BN classifiers are illustrated in Fig. 1.

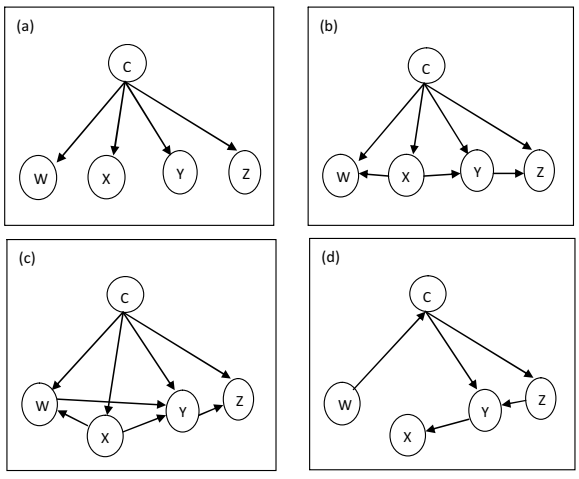

Fig.1 Different types of Bayesian classifiers are presented:(a)Naïve-Bayes (b) TAN (c) BAN (d) GBN

♦ Naïve-Bayes(NB). As mentioned earlier, this classifier is a simple and effective classification model. In NB classifier, the only parent node of all other nodes is the class node. Despite its simplicity, Naïve-Bayes has surprisingly outperformed several other more elaborated classifiers in many experiments, especially when the attributes are not strongly correlated-since in such cases the attribute-independency assumption is not so problematic.

♦ Tree Augmented Naïve-Bayes (TAN). This classifier is a simple and natural extension of Naïve-Bayes. Although Naïve-Bayes performance can be comparable with other classification methods, its attribute of independence assumption limits its real application. Extending its structure is a direct way to overcome the limitation of Naïve Bayes,since attribute dependencies can be explicitly represented by arcs. TAN is an extended tree-like naive Bayes in which the class node directly points to all attribute nodes and an attribute node can have only one parent from another attribute node.

♦ BN Augmented Naïve-Bayes (BAN). This type of classifier is more elaborated (and more computationally expensive to learn) than both Naïve- Bayes and TAN. In a BAN, there are no restrictions on the number of parents of a node. In other words, which further expanded the tree-like structure of TAN classifier and allowed the dependency relation between any two attribute nodes. Each node in BAN can have k parents besides the class node. Obviously, if k = 1, a BAN becomes a TAN.

♦ General Bayesian Network (GBN). GBN classifier is quite different from the others described earlier, since it treats the class variable node as an ordinary node (which can have both parent and child nodes) during the process of network-structure construction.

An excellent study of these algorithms was provided by Friedman et al. [20]. A comprehensive investigation and comparisons of various Bayesian classifiers was done by Cheng and Greiner [6,21]. A relatively recent survey on improving Naïve-Bayes for classification was found in [22].

III. Artificial fish swarm algorithm for learning BNC(AFS-BNC)

An artificial fish swarm algorithm is proposed to learn the structure of general Bayesian network classifier. In this algorithm, an unconstrained optimization problem is solved firstly, such technique is proposed in [14], its optimal solution is a undirect graph, which can be used to generate the initial population. Then, the artificial fish swarm algorithm for learning Bayesian network classifiers is presented. The details are given as below.

In our algorithm, each artificial fish represents a candidate BN. There exists a problem how to express a candidate BN.We use a representation method by Larranga etal[23] to express such BN. In their researches, the network structure(composed of n nodes) is represented by $n \times n$ adjacent matrix C. Each element c_{ij} in the matrix is defined as

$$c_{ij} = \begin{cases} 1 & \text{if node } i \text{ is a parent of node } j, \\ 0 & \text{otherwise.} \end{cases}$$

For example, the following Bayesian network(see Fig. 2) can be expressed by the adjacency matrix:

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

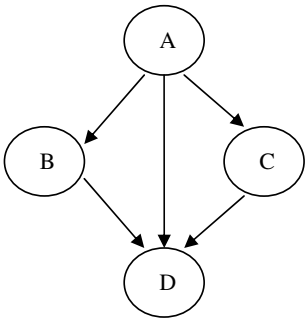

Fig.2 A network structure with four nodes

By flattening the matrix, the bit-string representation is obtained as following vector:

$$c_{11}c_{12}c_{13}\cdots c_{1n}c_{21}c_{22}c_{23}\cdots c_{2n}\cdots c_{n1}c_{n2}c_{n3}\cdots c_{nn}.$$

It will be used to represent the position of each artificial fish in our algorithm.

In AFS-GBN algorithm, let $x_i = (x_{i1}, x_{i2}, \cdots, x_{in})$ be the current position of artificial fish i, where n is the number of nodes. $Y_i = F(x_i)$ is the food consistence of AF, which can represent the fitness or objective function; $visual$ is the visual distance, δ is the crowed factor and $trynumber$ is the try number; N is the number of artificial fish. The distance of artificial fish x_i and x_j is defined as follows:

$$d_{ij} = \| x_i - x_j \| = \sum_{k=1}^{n} | x_{ik} - x_{jk} | .$$

In AFS-NBC algorithm, three behaviors are included:

(1) Preying behavior(AF-PREY): Let x_i be the AF current state and Y_i be the corresponding food consistence. By using the following formula to select a state x_j randomly within its visual distance:

$$x_{jk} = \begin{cases} \urcorner x_{ik}, & \text{randomly select a position of } x_i, \\ x_{ik}, & \text{other positions} \end{cases}$$

If $Y_i < Y_j$, then it goes forward a step in the direction of the Y_j, that is, let $x_i = x_j$.

Otherwise,select a state x_j randomly again and judge whether it satisfies the forward condition. If it can not satisfy after $trynumber$ times, it moves a step as follows:

$$x_{ik} = \begin{cases} \urcorner x_{ik}, & \text{randomly select a position of } x_i, \\ x_{ik}, & \text{other positions} \end{cases}$$

(2) Swarm behavior(AF-SWARM): Let x_i be the AF current state and Y_i be the corresponding food consistence. Taking x_i as a center to determine the number N_f of its sensing range. Let S_i be the collection of these artificial fish, which is defined as follows:

$$S_i = \{x_j \mid \| x_j - x_i \| \leq visual,\ j \neq i, j = 1, 2, \cdots, N\}.$$

Let x_c be the center position of S_i. If $N_f \geq 1$, then x_c can be determined by the following equation:

$$x_c = \frac{\sum_{j=1}^{N_f} x_j}{N_f}. \quad (4)$$

Because the state of each artificial fish is represented by binary, so, (4) can be represented as follows:

$$x_{ck} = \begin{cases} 1 & \rho \geq 0.5; \\ 0 & \rho < 0.5, \end{cases}$$

where $\rho = \frac{\sum_{j=1}^{N_f} x_{jk}}{N_f}$. If $Y_c > \delta * Y_i * N_f$, which means that companion center has more food and is not too crowded, and then goes forward a step in the direction of companion center x_c as follows:

$$x_{ik} = \begin{cases} x_{ck} & \text{randomly select a position of } x_c, \\ x_{ik}, & \text{other positions.} \end{cases}$$

If $N_f = 0$, the prey behavior is executed.

(3) Following Behavior(AF-FOLLOW): Let x_i be the AF current state, x_{max} be the companion with greatest food consistence within the visual range of current AF, and N_f be the number of companions within the visual range of current AF. If $Y_{max} > \delta * Y_i * N_f$, which means that x_{max} has higher food consistence and is not too crowded, and then goes forward a step in the direction of x_{max} by the following:

$$x_{ik} = \begin{cases} x_{max,k} & \text{randomly select a position in } x_{max}, \\ x_{ik}, & \text{other positions.} \end{cases}$$

(4) Bulletin board: Bulletin board is used to record the best AF state. After passing each evolution, the food consistence of the all fishes will be compared with bulletin board. If the AF's food consistence is higher, then replaces the bulletin board.

We should point out that the results of the three behaviors may be infeasible(i.e. the graphs corresponding to the positions obtained by the three behaviors may be cyclic graph), so, some strategies should be adopted to get rid of the ring).

Based on the above, we present our algorithm.

AFS-BNC Algorithm.

begin Procedure.

Step 0 Initialization

Generate N artificial fish randomly based on the undirect graph obtained by the method in [8]. Set the crowed factor δ, the visual distance $visual$, the maximum number of try $trynumber$ and the maximum number of iterations $maxiter$.

Step 1 Set $k = 1$;

Step 2 while $k \leq maxiter$

Step 3 For $i = 1 : N$

Step 4 Execute the following behavior. If $Y_j > \delta * Y_i * N_f$, goes forward a step in the direction of x_j, and updates the Bulletin board.

Step 5 else if execute the swarming behavior. If the condition is satisfied, the x_i moves a step forward, and updates the Bulletin board.

Step 6 else execute the preying behavior.

Step 7 If $m \leq trynumber$, the condition is satisfied, the x_i moves a step forward, and updates the Bulletin board; else executes the randomly behavior, and updates the Bulletin board.

Step 8 End for

Step 9 $k = k + 1$;

Step 10 End while

IV. Experimental evaluation

In this section, we are going to use some state-of-the art Bayesian classifiers like the NB classifier, the TAN classifier and the BNA classifier. We run our experiments on 12 data sets from the UCI repository of Machine Learning data sets [24], which include a wide range of domains and data characteristics. The description of the 12 data sets are given in Table 1. In our experiments, two useless attributes are manually deleted: the attribute "name" in the dataset Hayes-roth, and the attribute "animal name" in the dataset Zoo.

The experimental platform is a personal computer with Pentium 4, 3.06 GHz CPU, 512M memory,and Windows XP. The algorithm was implemented by Matlab7.0. In our method, the unconstrained optimization problem is solved by calling the function bintprog(•) in Matlab. These experimenal parameters are set as follows: the crowed factor $\delta = 0.11$, the visual distance $visual = 10$, the maximum try number $trynumber = 15$, the maximum iterations $maxiter = 20$.

Our implementation is based on the BayesNet Toolbox for Matlab [25], which provides source code to perform several operations on Bayesian networks. The aim of these experiments is to compare the performance of the proposed AFS-BNC with NB, TAN and BAN in terms of classifier accuracy. The accuracy of each model is based on the percentage of successful predictions on the test sets of each data set. In all experiments, the accuracy of each model on each data set are obtained via 10 runs of 5-fold cross validation. Runs with the various algorithms are carried out on the same training sets and evaluated on the same test sets. In particular, the cross-validation folds are the same for all the experiments on each data set.

Table 2 shows the accuracy (and standard deviation of accuracy) of each classifier on each data set. In each row, the best results of the four classifiers are displayed in bold. From our experiments, we can see that AFS-BNC is best in 7 cases. TAN and BAN are best in 4 and 1 cases, respectively. On the whole, AFS-BNC has a higher accuracy.

Figures 3, 4 and 5 show two scatter-plots comparing AFS-BNC with NB, TAN, and BAN, respectively. In the scatter plot, each point represents a data set, where the x coordinate of a point is the percentage of misclassifications according to NB or TAN or BAN, and the y coordinate is the percentage of misclassifications according to AFS-BNC. Thus, points below the diagonal line correspond to data sets on which AFS-BNC performs better. From Figures 3,4 and 5, we can see that AFS-BNC generally outperforms NB, TAN and BNA as is also demonstrated in Table 2. It provides strong evidence that AFS-BNC is performing well against the other three classifiers both in terms of accuracy and the percentage of misclassifications.

From the above discussion, it implies that AFS-BNC has a higher accuracy and a more simple graph structure. Thus, AFS-BNC is able to handle very large data sets and is a more promising classifier.

TABLE I
DESCRIPTIONS OF UCI DATASETS USED IN THE EXPERIMENTS

Dataset	♯Instance	♯Attributes	♯Classes	Missing value(N/Y)
Shuttle	15	6	2	N
Lenses	24	4	3	N
Zoo	101	17	7	N
Hayes-roth	132	5	3	N
Iris	150	4	3	N
Heart	267	22	2	N
Monk's	432	7	2	N
Vote	435	16	2	Y
Balance-scale	625	5	3	N
Tic-tac-toe	958	9	2	N
Car-evaluation	1728	6	4	N
Nursery	12960	8	5	N

TABLE II
THE DETAILED RESULTS ON ACCURACY AND STANDARD DEVIATION

Dataset	NB	TAN	BAN	AFS-BNC
Shuttle	83.52± 5.85	83.33± 7.03	84.43± 6.83	**86.667± 5.61**
Lenses	92.56± 2.44	**93.25± 2.85**	89.67± 3.16	92.63± 4.27
Zoo	93.46± 5.12	95.21± 4.78	94.34± 5.16	**96.32± 3.27**
Hayes-roth	81.82± 5.27	**87.34± 4.89**	84.58± 3.74	86.92± 2.18
Iris	94.13± 6.12	93.47± 6.65	93.81± 6.80	**97.14± 1.24**
Heart	82.41± 6.62	82.53± 7.01	**82.65± 7.03**	78.46± 4.36
Monk's	84.28± 1.14	83.29± 1.29	83.79± 1.23	**100.0± 0.00**
Vote	94.88± 3.57	90.37± 3.34	92.63± 3.98	**100.0± 0.00**
Balance-scale	90.91± 1.51	84.97± 2.44	87.94± 2.31	**91.61± 3.12**
Tic-tac-toe	70.11± 4.45	73.16± 4.47	75.74±4.83	**81.89± 1.06**
Car-evaluation	84.95± 2.74	**93.44± 1.89**	89.20±1.02	91.14± 3.28
Nursery	90.54± 1.14	**94.07± 1.29**	92.31±1.23	91.63± 2.69

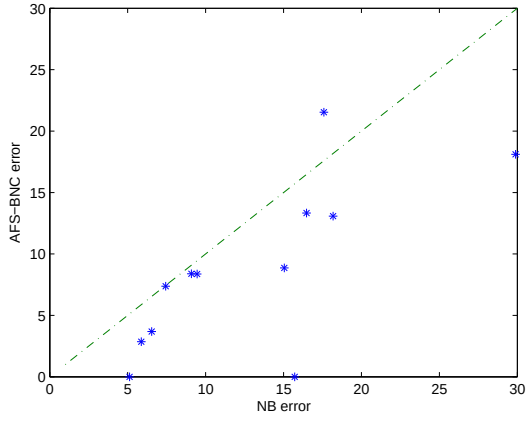

Fig.3 Relative errors of AFS-BNC and NB

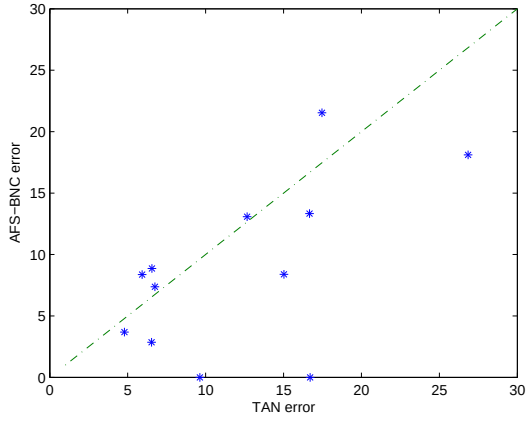

Fig.4 Relative errors of AFS-BNC and TAN

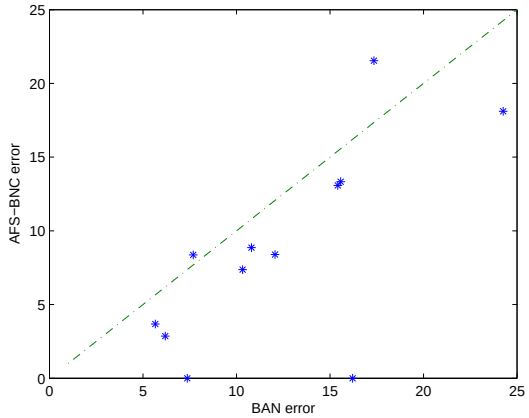

Fig.5 Relative errors of AFS-BNC and BAN

V. CONCLUSION

We have developed a new Bayesian network classifier AFS-NBC based on artificial fish swarm algorithm. The algorithm was tested on 12 data sets from UCI. The experimental results illustrate that the new algorithm has a better performance compared to NB, TAN and BAN.

REFERENCES

[1] J. Pearl, " Reasoning in Intelligent Systems: Networks of Plausible Inference," *Artificial Intelligence*, vol. 48, no. 1, pp. 117-124, 1991.

[2] O. Pietquin, T. Dutoit, "A probabilistic framework for dialog simulation and optimal strategy learning," *IEEE Transactions on Speech and Audio Processing*, vol. 14, no. 2, pp. 589-599, 2006.

[3] D. Nikovski, "Constructing Bayesian networks for medical diagnosis from incomplete and partially correct statistics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 4, pp. 509-516, 2000.

[4] P. Langley, W. Iba, K. Thompson," An Analysis of Bayesian Classifiers," in *10th National Conference on Artificial Intelligence (AAAI-92)*,AAAI Press, Palo Alto, CA, USA, 1992.

[5] M. G. Madden, "A New Bayesian Network Structure for Classification Tasks," in *Proceedings of 13th Irish Conference on Artificial Intelligence and Cognitive Science*, 2002.

[6] J. Cheng, R. Greiner, "Learning Bayesian Belief Network Classifiers: Algorithms and System," in *14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, Springer, London, UK, 2001.

[7] M. G. Madden, "On the Classification Performance of TAN and General Bayesian Networks ," in *Proceedings of AI-2008, the Twenty-eighth SGAI International Conference*, Cambridge, UK, 2008.

[8] K. M. Salama, A. A. Freitas, "Learning Bayesian network classifiers using ant colony optimization," *Swarm Intelligence*, vol. 7, no. 2-3, pp. 229-254, 2013.

[9] T. Sousa, A. Silva, A. Neves, "Particle swarm based data mining algorithms for classification tasks," *Parallel Computing*, Vol. 30, no. 5-6, pp.767-783, 2004.

[10] M. A. Shukran, Y. K. Chung, et al., "Artificial bee colony based data mining algorithms for classification tasks," *Modern Applied Science*, Vol. 5, no. 4, pp. 217-231, 2011.

[11] S. Dehuri, B. K. Nanda, S. B. Cho, " A Hybrid APSO-aided Learnable Bayesian Classifier," in *IICAI*, 2009.

[12] B. Heni,M. Afif, et al., " A new approach for Bayesian classifier learning structure via *K*2 algorithm," *Emerging Intelligent Computing Technology and Applications*, Vol. 304, pp.387-393, 2012.

[13] S. Moran, Y. Hey, K. C. Liu, "Choosing the best Bayesian classifier: an empirical study," *IAENG International Journal of Computer Science*, Vol. 36, no. 4, pp. 322-331, 2009.

[14] C. F. Wang, S. Y. Liu, M. M. Zhu, "Bayesian network learning algorithm based on unconstrained," *Journal of Systems Engineering and Electronics*,vol. 23, no. 5, pp.784-790, 2012.

[15] G. F. Cooper, E. Herskovits, "A Bayesian Method for the Induction of Probabilistic Networks from Data," *Machine Learning*,vol. 9, pp. 309-348, 1992.

[16] D. Heckerman, D. Geiger, D. M. Chickering, " Learning Bayesian Networks: the Combination of Knowledge and Statistical Data," *Machine Learning*,vol. 20, pp. 197-244, 1995.

[17] S. Yang, "Comparison of Score Metrics for Bayesian Network Learning, IEEE Transactions on Systems," *Man and Cybernetics, Part A: Systems and Humans*, vol. 32, pp. 419-428, 2002.
[18] W. Buntine, "Theory refinement on Bayesian networks," in *17th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Francisco, CA, USA, 1991.
[19] D. Heckerman, D. Geiger, D.M. Chickering, "Learning Bayesian Networks: the Combination of Knowledge and Statistical Data," *Machine Learning*, vol. 20, pp. 197-244,1995.
[20] N. Friedman, D. Geiger, M. Goldszmidt, " Bayesian Network Classifiers," *Machine Learning*, Vol. 29, pp. 131-161, 1997.
[21] J. Cheng, R. Greiner, "Comparing Bayesian Network Classifiers," in *15th Annual Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Francisco, CA,USA, 1999.
[22] L. Jiang, D. Wang, Z. Cai, X. Yan, "Survey of Improving Naive Bayes for Classification," in *International Conference on Advanced Data Mining and Applications*, Springer Heidelberg, LNCS 4632, 2007.
[23] P. Larranaga, I. Inza, J.L. Flores, "A guide to the literature on inferring genetic networks by probabilistic graphical models," Data Analysis and Visualization in Genomics and Proteomics, John Wiley and Sons, Ltd., 2005.
[24] A. Frank, A. Asuncion, UCI Machine Learning Repository, University of California, School of Information and Computer Science,Irvine,Calif,USA,2010,http://archive.ics.uci.edu/ml/.
[25] K. P. Murphy, BaysNet Toolbox For Matlab, 2007, https://code.google.com/p/bnt/.

Chun-Feng Wang was born in 1978. He received his M.S. degree in 2006 from Henan Normal University, and received his Ph.D. degree in 2012 from Xidian University. His research interests include Bayesian network learning, optimization theory and its applications.

Kui Liu 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, Xian, China, in 2008, and the Ph.D. degree in applied mathematics from Xidian University, Xian, China.

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