Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A Granular Computing Approach to Improve Large Attributes Learning*

Fengming M. Chang†
Department of Information Science and Applications
Asia University
Wufeng, Taichung 41354, Taiwan
paperss@gmail.com

Chien-Chung Chan
Department of Computer Science
University of Akron
Akron, OH 44325-4003,USA
chan@uakron.edu

Abstract **– Based on the concept of granular computing, this article proposes a novel Boolean Conversion (BC) method to reduce data attribute number for the purpose of improving the efficiency of learning in artificial intelligence. Data with large amount of attributes usually cause a system freezes or shuts down. The proposed method combines large amount attributes to smaller number ones by the way of Boolean method. Three data sets are used to compare the learning accuracies and efficiencies by Bayesian networks (BN), C4.5 decision tree, support vector machine (SVM), artificial neural network (ANN), fuzzy neural network (FNN, neuro-fuzzy), and Mega-fuzzification learning methods. Results indicate that the proposed BC method can improve the efficiency of machine learning and the accuracy is not worse.**

I. INTRODUCTION

Artificial intelligence (AI) methods have been widely used in many fields recently for data prediction [1-14]. In many cases, data used for AI researches have small number of attributes. However, in the real world, data sets usually have large number of attributes. This condition causes large calculation work problems in performing AI computation and computers are easy to be hold or shut down. In different kinks of learning methods, artificial neural network (ANN), fuzzy neural network (FNN, neuro-fuzzy), and FNN based mega-fuzzification [9, 15-16] need large computation works and are easy to cause a computer freezes. In our experiment, the number of data attributes exceeds 6 causes FNN and mega-fuzzification fail to perform. Therefore, the cost of acquiring exact information from large number of attributes may be too high, and coarse-grained information may serve the need and reduce cost [17].

In this paper, a granular computing based Boolean Conversion (BC) method was proposed to reduce the cost in terms of computing time in the context of Adaptive Network based Fuzzy Inference Systems (ANFIS) learning, a neuro-fuzzy learning proposed by Jang [18]. ANFIS performs a fuzzy inference system by adaptive backpropagation learning with given fuzzy membership input variables. In the experiment of [9], the performance of ANFIS may deteriorate due to heavy computation works by a large number of data attributes. As observed in the study of Chang and Chan [10, 17], ANFIS might take a long time to complete a learning task when the number of attributes exceeds 6, and fail to run to the end for some data sets. However, from the view point of machine learning, 6 is not a large number. The purpose of the proposed BC method is to solve above problem in FNN and FNN based mega-fuzzification. Nevertheless, other machine learning methods are also used to compare the results of BC. Such methods include Bayesian networks (BN), C4.5 decision tree, support vector machine (SVM), and ANN learning methods.

The rest of the paper is organized as follows. Section II reviews the related works. The proposed BC method is described in Section III. The computational results by three cases are presented in Section IV. Finally, conclusions are given in Section V, and it is followed by references.

II. RELATED WORKS

Data attribute reduction is an important way to improve the efficiency of AI learning. Early related work was done by Shen and Chouchoulas, who proposed a Rough Set Attribute Reduction (RSAR) method to remove redundant input attributes for discrete values from complex systems. However, RSAR still lacks efficiency although it can reduct attributes [18]. The other study is that Beynon introduced an “approximate reducts” concept and proposed a Variable Precision Rough Sets (VPRS) model to find out the smallest set of attributes [19]. Later, Hsu et al. applied VPRS model for mobile phone test procedure [20]. Inbarani et al. also applied VPRS for feature selection of web usage mining [21]. In addition, Ang and Quek did not reduce data attriubute but reduce fuzzy rules by combined rough set and neuro-fuzzy learning [22].

III. BOOLEAN CONVERSION

A PUSTAK data set that was taken from the Rose2 software sample database [23] is used to explain the proposed BC method first. PUSTAK has 10 input and 1 output attributes. It can not be performed well in FNN and mega-fuzzification methods. Values of its 11 attributes are integers. Values of the first input to the ninth input attributes are from 1 to 6, of the tenth input are from 1 to 5, and values of output are 1 and 2.

* This work is partially supported by NSC 96-2416-H-468-006-MY2.
† Corresponding author.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

The process of the BC method is simple. Each decimal number can be transferred into a Boolean number one on one mapping. On the top of the FIG. 1, five decimal numbers, 5, 1, 1, 2, and 2, are transferred into five Boolean numbers accordingly. Considering the maximum value of each attribute, Boolean number for decimal number 1 should be 001, 2 should be 010, and so on. So that the number of bit in Boolean number for each attribute is fixed. Next, the five Boolean numbers are physically combined to be a unique Boolean number as shown in the middle part of the FIG. 1. Each original Boolean number occupies its own digital position in the combined Boolean number format without mixing with other numbers. After that, for the convenience of calculation in the real world, this combined Boolean number is transferred to a decimal number. In the above process, the five input values are combined into a unique value and the input attributes are reduced. The reason for not combining the decimal number directly, such as combine 5, 1, 1, 2, 2 to be 51122 is because that 51122 is bigger than 21074, the result of BC. Smaller number is easier for calculation.

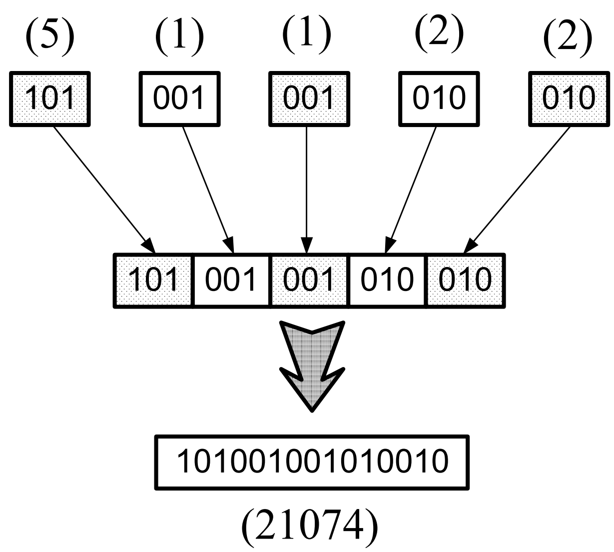

FIG. 1. THE PROCESS OF THE BOOLEAN CONVERSION.

With FIG. 1 as an example, there are 5 inputs were converted into a single new input. First, the original 5 inputs {5, 1, 1, 2, 2} are converted into a Boolean digit number that is {101, 001, 001, 010, 010}. Second, these 5 Boolean digit numbers are physically combined into one Boolean digit number: 101001001010010. The corresponding decimal number is 21074. It can be expressed by the binary system as:

$$
\begin{aligned}
&\{\mathbf{101, 001, 001, 010, 010}\} \\
&\rightarrow 101001001010010 \\
&= \mathbf{101} * 1000000000000 \\
&+ \mathbf{001} * 1000000000 \\
&+ \mathbf{001} * 1000000 \\
&+ \mathbf{010} * 1000 \\
&+ \mathbf{010} * 1
\end{aligned}
$$

It could be a Boolean weight expressed by binary system as:

$$\boldsymbol{B} = [1000000000000 \quad 1000000000 \quad 1000000 \quad 1000 \quad 1]$$

or expressed by decimal system:

$$\{\mathbf{5, 1, 1, 2, 2}\} \rightarrow 21074 = \mathbf{5} * 2^{12} + \mathbf{1} * 2^{9} + \mathbf{1} * 2^{6} + \mathbf{2} * 2^{3} + \mathbf{2} * 2^{0}$$

and the binary weight vector is

$$\boldsymbol{B} = [2^{12} \quad 2^{9} \quad 2^{6} \quad 2^{3} \quad 2^{0}]$$

power of 2 is determined by the data maximum domain. For example, if the value range of the second attribute is from 1 to 6, the maximum domain is 6, then

$$6(decimal) = 110(binary) = 2^2 + 2^1(decimal) \leq 2^3(decimal)$$

2^3 means the number of bits the second attribute needs is 3, $bits(2) = 3$. As well as the third to the fifth attributes, each one needs 3 also. This is a special condition in this case that all the attributes need 3, but each attribute may not need the same number of bits in other cases. The power of the second attribute is the bit number plus the power of the third attribute or the total bit number from the third to the fifth inputs. It is presented as:

$$power(i) = bits(i+1) + power(i+1) = \sum_{m=i+1}^{I} bits(m)$$

$$1 \leq i \leq (I-1), \quad power(I) = 0,$$

where I is the total number of the input i.

IV. COMPUTATIONAL RESULTS

A. Case 1: PUSTAK

PUSTAK data that is described in section III is used by the proposed method in this subsection. TABLE I shows one record of the data. There are 10 input and one output attributes in the data. The 1st to the 5th attributes are converted into a new input attribute, the 6th to the 10th attributes are combined into another new input by BC. Therefore there are only two new input attributes. As shown in TABLE I, the new input record is {21074, 8780}.

After all attributes are converted using BC, the data are tested and compared using BN, C4.5, SVM, ANN, FNN, and Mega-fuzzification methods with 10-folds cross-validation testing. Each fold are used as testing data in turn and the remaining total of 9 folds data are used as training data. The results are presented in TABLE II. Without using BC, FNN and Mega-fuzzification fail to perform. After applying BC, it can easily perform machine learning using FNN and Mega-fuzzification methods. The prediction accuracies after using BC are a little lower than without using it in this case, but learning time decreases. FIG. 2 compares the prediction accuracies under different learning methods, and FIG. 3 compares the leaning time needed.

TABLE I
AN EXPLANATION OF CONVERTING 10 ATTRIBUTES TO 2 NEW DECIMAL ATTRIBUTES.

	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Original decimal value	5	1	1	2	2	2	1	1	1	4
Converse to Boolean value	101	001	001	010	010	010	001	001	001	100
Combine to two Boolean value	101001001010010					10001001001100				
Converse to two decimal value	21074					8780				

TABLE II
THE COMPARISON OF PUSTAK DATA.

	Method	Bayesian	C4.5	SVM	ANN	FNN	Mega-fuzzification
Non-BC	Accuracy	86.11%	94.91%	90.74%	89.81%	Fail to perform	Fail to perform
	Time(sec)	0.05	0.03	0.41	1.06		
BC	Accuracy	83.80%	88.43%	85.19%	87.96%	79%	82%
	Time(sec)	0	0	0.16	0.3	4.2	4.2

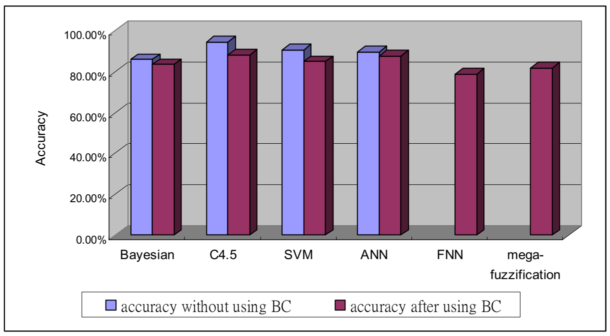

FIG. 2. THE ACCURACY COMPARISON BEFORE AND AFTER USING BC MEHTOD BY SIX MEHTODS FOR PUSTAK DATA.

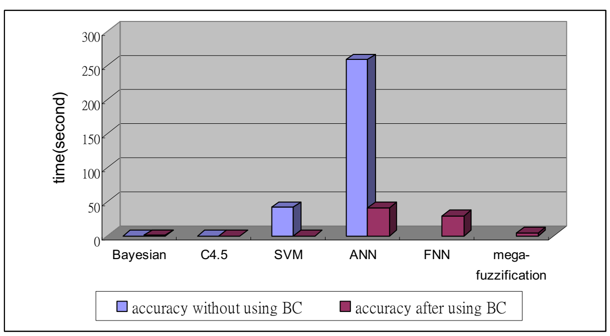

FIG. 3. THE LEARNING TIME COMPARISON BEFORE AND AFTER USING BC MEHTOD BY SIX MEHTODS FOR PUSTAK DATA.

B. Case 2: Heart disease

The heart disease data are from Cleveland Hospital provided by Dr. Detrano (see UCI Machine Learning Repository [24]). There are in total 303 patients' records in the data. Each record of the data has 13 inputs and one output attribute. Some items' values in the data are missing, we got 296 records of data.

The total 14 attributes include 8 symbolic and 6 numeric items. For the convenience to some learning methods, symbolic attributes are transferred to numeric before learning. The 14 attributes with their values are list as the following: Age, Sex (1,0), Chest pain type (angina, abnang, notang, asympt) or (1-4), Trestbps (resting blood pres), Cholesterol, Fasting blood sugar < 120 (true or false) or (1,0), Resting ecg (norm, abn, hyper) or (0,1,2), Max heart rate, Exercise induced angina (true or false) or (1,0), Oldpeak, Slope (up, flat, down) or (1,2,3), Number of vessels colored, Thal (norm, fixed, rever) or (3,6,7), Either healthy (buff) or with heart-disease (sick) or (0 is healthy, 1,2,3,4 is sick).

There are two kinds of output classifications used. One is to classify output into healthy and sick, two values, the other is to classify output into healthy and four levels of sick, total of five values. It seems that to classify output only healthy and sick is more reasonable and accurate than classify into five values in our experiments later.

TABLE III
THE COMPARISON OF HEART DISEASE DATA BY CLASSIFYING OUTPUT TO 1 HEALTHY AND 4 SICK DEGREES.

	Method	Bayesian	C4.5	SVM	ANN	FNN	Mega-fuzzification
Non-BC	Accuracy	56.09 %	50.68%	55.46%	52.70%	Fail to perform	Fail to perform
	Time(sec)	0.01	0.06	41.7	257.8		
BC	Accuracy	53.38%	54.73%	53.38%	55.41%	53%	55%
	Time(sec)	0.01	0.02	41.5	29.5	4.7	4.8

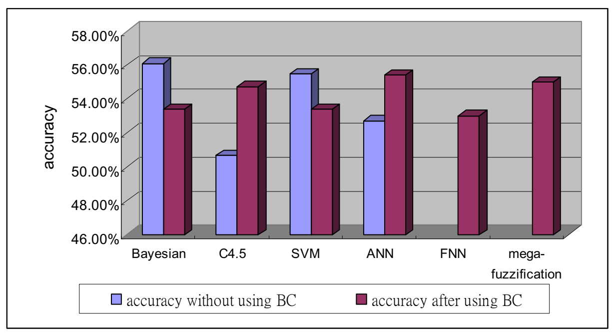

FIG. 4. THE ACCURACY COMPARISON FOR HEART DISEASE DATA BY CLASSIFYING OUTPUT TO 1 HEALTHY AND 4 SICK DEGREES.

When classifying output to 1 healthy and 4 sick degrees, the learning accuracies and time are presented in TABLE III and FIG. 4. In this case, not all learning accuracies after using BC are lower than before. When classifying output to healthy and sick two values, the learning accuracies and time are presented in TABLE IV and FIG. 5. In this case, all learning accuracies after using BC are higher than before. Such results are different from the PUSTAK case.

Still, before using BC method, learning in FNN and mega-fuzzification methods fails in heart disease data.

TABLE IV
THE COMPARISON OF HEART DISEASE DATA BY CLASSIFYING OUTPUT TO HEALTHY AND SICK TWO VALUES.

	Method	Bayesian	C4.5	SVM	ANN	FNN	Mega-fuzzification
Non-BC	Accuracy	62.09%	58.65%	57.03%	57.03%	Fail to perform	Fail to perform
	Time(sec)	0.02	0.07	0.76	201.3		
BC	Accuracy	73.31%	71.96%	69.59%	67.91%	73.05%	74.13%
	Time(sec)	0.01	0.01	0.42	9.9	4.3	4.4

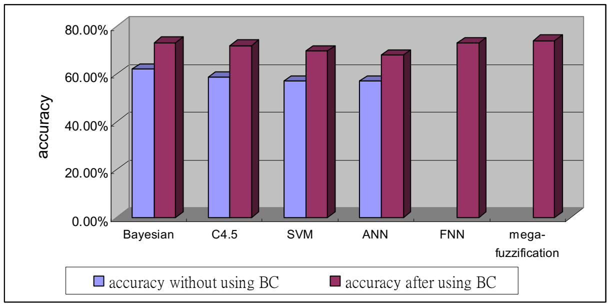

FIG. 5. THE ACCURACY COMPARISON FOR HEART DISEASE DATA BY CLASSIFYING OUTPUT TO HEALTHY AND SICK TWO VALUES.

C. *Case 3: Hayes-Roth*

Hayes-Roth data were created by Barbara and Frederick Hayes-Roth (see UCI Machine Learning Repository [25]) which has 132 instances, 4 inputs and 1 output attributes. Because the number of attributes is not large in this case, we can compare the learning accuracies of FNN and mega-fuzzificatioin with and without using BC. TABLE V shows the results. In this case, FNN and mega-fuzzification can be performed well before using BC. All the accuracies after using BC are a little lower than before. The learning accuracies are also compared in FIG. 6.

TABLE V
THE COMPARISON OF HAYES-ROTH DATA.

	Method	Bayesian	C4.5	SVM	ANN	FNN	Mega-fuzzification
Non-BC	Accuracy	74.24%	80.30%	53.79%	74.24%	69%	70%
	Time(sec)	0	0	0.31	0.3	36	33
BC	Accuracy	54.55%	68.94%	51.52%	56.06%	63%	65%
	Time(sec)	0	0	0.29	0.19	3	3

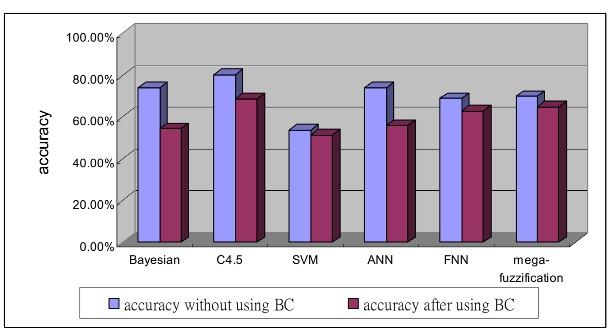

FIG. 6. THE ACCURACY COMPARISON BEFORE AND AFTER USING BC MEHTOD BY SIX MEHTODS FOR HAYES-ROTH DATA.

V. CONCLUSIONS

When number of data attributes is large, it becomes an obstacle in machine leaning. Some leaning methods need a large of computation works, and large amount attributes causes a system easy to freezes. On the other hand, some leaning systems have their limits and can not allow data have too many attributes. Therefore, this study proposed a granular computing based Boolean Conversion method to reduce data attributes by a binary transfer process. After attributes are combined and reduced, with three data sets, we compare the learning results using BN, C4.5, SVM, ANN, neuro-fuzzy, and Mega-fuzzification learning methods. In some cases, learning accuracies after using BC are a little lower than before, some cases have higher accuracies, and some cases have both higher and lower accuracies. In general, the learning accuracy after applying BC is not worse. In addition, leaning time is shortened after BC is used. Facing the problem of "fail to perform" in neuro-fuzzy, the proposed BC method indeed solves the problem of data have large attributes in learning in brief.

ACKNOWLEDGMENT

Thanks are due to the support in part by the National Science Council of Taiwan under Grant No. NSC 96-2416-H-468-006-MY2.

REFERENCES

[1] Y.Y. Yao, "Granular computing: basic issues and possible solutions," Proceedings of the 5th Joint Conference on Information Sciences, 1999, pp. 186 – 189.

[2] L. Polkowski and A. Skowron, "Towards adaptive calculus of granules," *Proceedings of 1998 IEEE International Conference on Fuzzy Systems*, pp. 111 – 116.

[3] T.Y. Lin, "Granular computing on binary relations I: data mining and neighborhood systems, II: Rough set representations and belief functions," in L. Polkowski and A. Skowron eds., *Rough sets in knowledge discovery 1*. Heidelberg, Physica-Verlag, 1998, pp. 107 – 140.

[4] Y.Y. Yao, "Granular computing using neighborhood systems," in R. Roy, T. Furuhashi , and P.K. Chawdhry (eds.) *Advances in Soft Computing: Engineering Design and Manufacturing*, Springer-Verlag, London, 1999, pp. 539 – 553.

[5] T.Y. Lin, "Data mining: granular computing approach," *Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, 1999, pp. 24 – 33.

[6] A. Skowron and J. Stepaniuk, "Information granules: towards foundations of granular computing," *International Journal of Intelligent Systems*, Vol. 16, 57 – 85, 2001.

[7] Y.Y. Yao, "Information granulation and rough set approximation," *International Journal of Intelligent Systems*, Vol. 16, 87 – 104, 2001.

[8] J.-S. R. Jang, "ANFIS: Adaptive-Network-based Fuzzy Inference Systems," *IEEE Transactions on System, Man, and Cybernetics*, vol. 23, no.3, pp. 665-685, 1993.

[9] D. C. Li, C. Wu, and F. M. Chang, "Using data-fuzzifying technology in small data set learning to improve FMS scheduling accuracy," *International Journal of Advanced Manufacturing Technology*, vol. 27, no. 3-4, pp. 321-328, 2005.

[10] F. M. Chang, and C. C. Chan, "Improve Neuro-Fuzzy Learning by Attribute Reduction," *The 27th Annual Meeting of the North American Fuzzy Information Processing Society,* The Rockefeller University**,** NY, USA, May 18-21, 2008.

[11] B. Predki, R. Slowinski, J. Stefanowski, R. Susmaga, and Sz. Wilk, "ROSE - Software Implementation of the Rough Set Theory," In: L. Polkowski, A. Skowron, eds, "Rough Sets and Current Trends in Computing," *Lecture Notes in Artificial Intelligence*, vol. 1424, pp. 605-608,. 1998.

[12] B. Predki and Sz.Wilk, "Rough Set Based Data Exploration Using ROSE System," In: Z. W. Ras, A. Skowron, eds, "Foundations of Intelligent Systems," *Lecture Notes in Artificial Intelligence*, vol. 1609, pp.172-180, 1999.

[13] A. Øhrn and J. Komorowski, "ROSETTA: a rough set toolkit for analysis of data," *Proc. Third International Joint Conference on Information Sciences*, Vol. 3, pp. 403 - 407, Durham, NC, March 1997.

[14] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer, 1991.

[15] D. C. Li, C. Wu, and F. M. Chang, "Using data continualization and expansion to improve small data set learning accuracy for early flexible manufacturing system (FMS) scheduling," *International Journal of Production Research*, vol. 44, No. 21, pp.4491-4509, 2006.

[16] F. M. Chang, "An intelligent method for knowledge derived from limited data," *Proceedings - 2005 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1, Waikoloa, The Big Island, Hawaii, USA, Oct. 10-12, pp.566-571, 2005.

[17] F. M. Chang, and C. C. Chan, "A Granular Computing Approach to Data Engineering," *Proceedings - 2008 IEEE International Conference on Systems, Man, and Cybernetics*, Singapore, Oct. 12-15, pp.2753-2758, 2008

[18] S. Qiang, and C. Alexios, "A modular approach to generating fuzzy rules with reduced attributes for the monitoring of complex systems," *Engineering Applications of Artificial Intelligence*, vol. 13, No. 3, pp.263-278, 2000.

[19] M. Beynon, "Reducts within the variable precision rough set model: A further investigation," *European Journal of Operational Research*, vol. 134, pp.592-605, 2001.

[20] J. H. Hsu, T. L. Chiang, and H. C. Wang, "VPRS model for mobile phone test procedure," *Journal of the Chinese Institute of Industrial Engineers*, vol. 23, no. 4, pp.345-355, 2006.

[21] H. H. Inbarani, K. Thangavel, and A. Pethalakshmi, "Rough set based Feature Selection for Web Usage Mining," *International Conference on Computational Intelligence and Multimedia Applications*, pp.33-38, 2007.

[22] K. K. Ang, and C. Quek, "Stock Trading Using RSPOP: A Novel Rough Set-Based Neuro-Fuzzy Approach," *IEEE Transactions on Neural Network*, vol. 17, no. 5, pp.1301-1315, 2006.

[23] Laboratory of Intelligent Decision Support Systems, Poznan University of Technology, http://www-idss.cs.put.poznan.pl/site/rose.html

[24] UCI Machine Learning Repository, http://mlearn.ics.uci.edu/MLRepository.html