

A Brief Review of Automatic Rule Generation

Author: Shin-Jye Lee¹; Ching-Hsun Tseng²

Affiliation: Institute of Management of Technology, National Chiao Tung University, Taiwan¹;

Department of Computer Science, University of Manchester, United Kingdom²

E-mail: camhero@gmail.com¹; hank131415go61@gmail.com²

DOI: 10.26821/IJSHRE.9.6.2021.9618

ABSTRACT

Automatic rule generation can be regarded as rule generation techniques, and it can be decomposed into rule induction and rule-based optimization. As the aspect of the size of the system, it just need to work rule induction in a simple or a small system with a few variables in the relative database without processing the step of rule-based optimization, because the performance of rule-based optimization is almost achieved by the step of rule induction in a simple or a small system. So far as the size of work space, the family of shared partitions is suitable to work in a small size work space with a good convergence, and the family of clustering is appropriate to work in a large size work space with a good convergence. However, the family of shared partitions is easy to make the problem of curse of dimensionality when dealing with large systems.

Keywords: Automatic Rule Generation, Partition

1. INTRODUCTION

Automatic rule generation can be regarded as rule generation techniques, So far as automatic rule generation is concerned, it can be decomposed into two main steps: rule induction and rule-based optimization [1]. As the aspect of the size of the system, it just need to work rule induction in a simple or a small system with a few variables in the relative database without processing the step of rule-based optimization, because the performance of rule-based optimization is almost

achieved by the step of rule induction in a simple or a small system. However, these two steps are both required for a complex or a big system with a number of variables in the relative database. It's easier to interpret a variable in a complex system with huge numbers of data by selecting substantial variables in the step of rule induction, and then the step of rule-based optimization will optimize the rule-base by removing useless rules in the relative database [1]. Definitely, variable selection and rule reduction are essential in the automatic rule generation process. Based on the family of methods, there are three types of methodologies trying to perform rule generation effectively, including shared partitions, clustering and hybrid methods [1].

2. SHARED PARTITIONS

It's very common to generate the partition by dividing each input variable domain into a given number intervals and whose boundary can be calculated by several approaches without any specified meaning. In the family of shared partitions, there are three approaches, including all the possible rules, one rule per pair and decision tree [1].

(1) All the possible rules [1]

In a multi-input-single-output system (MISO), *Ishibuchi et al.* [2] assume the input and output spaces to be $[0,1]^P$ and $[0,1]$. For the i th input variable x_i , its

domain interval is equally divided into K_i fuzzy sets labeled as $A_i^1, A_i^2, \dots, A_i^{K_i}$, so the total number of rules for a p -input system is $K_1 \times K_2 \times \dots \times K_p$. Also, the membership function can be defined as follows:

$$\mu_{ij}(x) = \max \left\{ 1 - \frac{|x - a_j^{K_i}|}{b^{K_i}}, 0 \right\} \quad (1)$$

where $a_j^{K_i} = \frac{j-1}{K_i-1}, j = 1, \dots, K_i, b^{K_i} = \frac{1}{K_i-1}$.

Nozaki *et al.* [3] propose a simple heuristic method to calculate the rule conclusions. For rule i , the conclusion is being written b_i , and which can be calculated as follows:

$$b_i = \frac{\sum_{j=1}^n w_i(j) * y(j)}{\sum_{j=1}^n w_i(j)} \quad (2)$$

where $y(j)$ presents the j pair observed output and $w_i(j)$ presents the i rule firestrength for the j pair, and the j pair inferred output can be calculated as follows:

$$\hat{y}(j) = \frac{\sum_{i=1}^r w_i(j) * b_i}{\sum_{i=1}^r w_i(j)} \quad (3)$$

In order to initialize the corresponding conclusions, Glorennec [4] proposes a diffusion procedure. S is the set of rules and whose conclusions have already been initialized. N_i is the neighborhood of rule i , and has been already defined by the sets of rules. Therefore, the N_{S_i} neighborhood can be defined as follows [1]:

$$N_{S_i} = \begin{cases} \{i\}, i \in S \\ N_i, otherwise \end{cases} \quad (4)$$

Based on the definition of (4), each rule has at most two neighbors in each input space dimension, and it can be defined as $|N_i| \leq 2^p$. Therefore, the diffusion can be calculated based on the following series $g^{(n)}$ [1]:

$$g^{(0)}(i) = \begin{cases} b_i, i \in S \\ 0, otherwise \end{cases} \quad (5)$$

$$g^{(n+1)}(i) = \frac{1}{2} \sup_{j \in N_{S_i}} g^{(n)}(j) + \frac{1}{2} \inf_{j \in N_{S_i}} g^{(n)}(j) \quad (6)$$

The series converges when n goes toward infinity and the diffusion procedure is stable.

(2) Only one rule per data pair [1]

Wang and Mendel [5] proposed a concept about one rule per data pair, and the number of rules is limited by the number of training pairs. Also, the procedure can be described as follows [1]:

- Each variable of the input space is automatically divided into a defined number of membership fuzzy sets.
- One fuzzy rule is generated for each data pair, the i th pair one can be written as follows:

If x_1 is A_1^i and x_2 is $A_2^i \dots$ and x_p is A_p^i then y is C^i . (7)

where A_p^i is the input variable p from pair i , C^i is the output variable from pair i .

- A degree is assigned to each rule, and it's equal to the rule firestrength for the consider pair. The

degree is the product of the firestrength by the confidence level.

- Expert rules are allowed. The and rules induced from data may be combined with or rules given by experts. It means or and and type rules are equally managed.
- The output is computed through the centroid defuzzification.

(3) Decision Trees [1]

Quinlan [6] proposed the concept of the decision trees in 1986. The purpose of the decision trees is to design paths leading to pure leaves with each leaf corresponding to an incomplete rule, and the tree represents a subspace of all the possible rules. Basically, the tree induction is an iterative process. A node corresponds to an input variable and generates a number of subnodes equal to the number of fuzzy sets of the selected variable. The process is repeated until all leaves are pure. The tree can be regarded as a source of a message. Therefore, the rule associated to a given node b can be written as follows [1]:

If x_{i_1} is $A_{i_1}^{j_1}$ and x_{i_2} is $A_{i_2}^{j_2}$... and x_{i_n} is $A_{i_n}^{j_n}$ then y is C_b . (8)

where $A_{i_1}^{j_1}$ corresponds to the first node of the path starting from the root and leading to the node b , and C_b is the most represented class in node b .

3. FUZZY CLUSTERING

Fuzzy Clustering algorithms can be regarded as a well-identified family of rule induction techniques, which organize and categorize data into homogeneous groups. Basically, each partition presents a rule associated to each cluster [1]. In 1974, Dunn [7] proposed a method of fuzzy clustering called Fuzzy C-

Means Clustering., and it is a method of clustering which allows one piece of data to belong to two or more clusters. Also, it is based on minimization of the following objective function [8]:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2, \quad 1 \leq m < \infty \quad (9)$$

where m is any real number greater than 1.0, u_{ij} is the degree of membership of x_i in the cluster j , x_i is the i th of d -dimensional measured data, c_j is the d -dimension centre of the cluster, and $\|*\|$ is any norm expressing the similarity between any measured data and the centre.

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (10)$$

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m} \quad (11)$$

This iteration keeps performing until the algorithm converges. In other words, when $\max_{ij} \left\{ |u_{ij}^{(k+1)} - u_{ij}^{(k)}| \right\} < \varepsilon$, where ε is a termination criterion between 0 and 1.0, whereas k are the iteration steps, it means the algorithm converges and this procedure converges to a local minimum or a saddle point of J_m [8].

4. HYBRID METHODS

Hybrid methods integrate a variety of theories, and try to perform the advantages of each method respectively from different theories. Basically, Neural Networks and Genetic Algorithm (GA) are widely worked in Hybrid Methods recently. The feature of

Neural Networks can bring their learning algorithms and numerical accuracy to fuzzy system without paying much attention to the semantic. The feature of Genetic Algorithms is more likely to find a global optimum and may optimize both the structure and the parameters of the corresponding fuzzy system. Moreover, both these methods can bring an excellent performance when the expert knowledge is not available in the fuzzy system [1].

5. CONCLUSION

So far as the size of work space, the family of shared partitions is suitable to work in a small size work space with a good convergence, and the family of clustering is appropriate to work in a large size work space with a good convergence. However, the family of shared partitions is easy to make the problem of curse of dimensionality when dealing with large systems. Also, the rule base completeness is not guaranteed though a weak convergence. In the family of clustering, the induced rule legibility gets worse as the work space size gets larger. Also, the relationship can be illustrated as Fig. 1. [1].

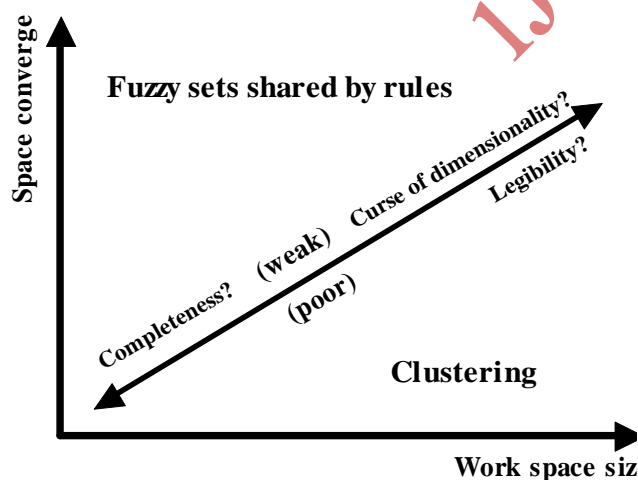


Fig. 1. The relationship and the comparison between the family of shared partitions and that of clustering [1]

6. REFERENCE

[1] Guillaume, S., (2001), Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented

Review, IEEE Transactions on Fuzzy Systems, Vol. 9, No. 3, pp. 426-443.

- [2] Ishibuchi, H., Nozaki, K., Tanaka, H., Hosaka, Y., and Matsuda, M., (1994), Empirical study on learning in fuzzy systems by rice test analysis, Fuzzy Sets Syst., vol. 64, pp. 129-144.
- [3] Nozaki, K., Ishibuchi, H., and Tanaka, H., (1997), A simple but powerful heuristic method for generating fuzzy rules from numerical data, Fuzzy Sets Syst., vol. 86, pp. 251-270.
- [4] Glorennec, P. Y., (1999), Algorithmes d'apprentissage pour systems d'inference floue, Paris, France: Hermes.
- [5] Wang, L. X. and Mendel, J. M., (1992), Generating fuzzy rules by learning from examples, IEEE Trans. Syst., Man, Cybern., vol. 22, pp.1414-1427.
- [6] Quinlan, J. R., (1986), Induction of decision trees, Mach. Learn., vol. 1, pp. 81-106.
- [7] Dunn, J. C., (1973), A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters, J. Cybern., vol. 3, no. 3, pp. 32-57.
- [8] Matteucci, M., (2003), Fuzzy C-Means Clustering, http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/cmeans.html, access: 30/04/06