Short Paper

A Genetics-Based Approach to Knowledge Integration and Refinement

CHING-HUNG WANG, TZUNG-PEI HONG[*] AND SHIAN-SHYONG TSENG[+]
Chunghwa Telecommunication Laboratories
Chungli, Taiwan 326, R.O.C.
E-mail: amidofu@cht.com.tw
[*]*Department of Electrical Engineering*
Nation University of Kaohsiung
Kaohsiung, Taiwan 840, R.O.C.
E-mail: tphong@nuk.edu.tw
[+]*Institute of Computer and Information Science*
National Chiao Tung University
Hsinchu, Taiwan 300, R.O.C.
E-mail: sstseng@cis.nctu.edu.tw

In this paper, we propose a genetics-based knowledge integration approach to integrate multiple rule sets into a central rule set. The proposed approach consists of two phases: knowledge encoding and knowledge integrating. In the encoding phase, each knowledge input is translated and expressed as a rule set, and then encoded as a bit string. The combined bit strings form an initial *knowledge population*, which is then ready for integrating. In the knowledge integration phase, a genetic algorithm generates an optimal or nearly optimal rule set from these initial knowledge inputs. Furthermore, a rule-refinement scheme is proposed to refine inference rules via interaction with the environment. Experiments on diagnosing brain tumors were carried out to compare the accuracy of a rule set generated by the proposed approach with that of initial rule sets derived from different groups of experts or induced by means of various machine learning techniques. Results show that the rule set derived using the proposed approach is much more accurate than each initial rule set on its own.

Keywords: brain tumor, expert system, genetic algorithm, knowledge integration, knowledge refinement

1. INTRODUCTION

Recently, Wang *et al*. proposed several GA-based knowledge integration strategies to automatically integrate multiple rule sets in a distributed-knowledge environment [7, 10-13]. Also, a self-integrating knowledge-based brain tumor diagnostic system based on these strategies was successfully developed [9]. In this paper, we propose a genet-

Received May 7, 1998; revised August 5, 1998; accepted October 9, 1998.
Communicated by Jieh Hsiang.

ics-based knowledge integration and refinement approach which operates at the rule-set level to effectively integrate multiple rule sets into one centralized knowledge base. The proposed approach takes less processing time than do those in [7]. It does not need to apply any domain-specific genetic operators to solve misclassification and contradiction problems. Instead, it used a refinement approach to effectively solve them. Also, domain experts need not intervene in the integration process since the work is done by computers.

Experiments on diagnosing brain tumors will be described. Results show that the knowledge base derived using our approach is much more accurate than each initial rule set on its own.

The remainder of this paper is organized as follows. The genetics-based knowledge-integration approach is proposed in Section 2. A rule-refinement scheme is proposed in Section 3. Experiments on diagnosing brain tumors are reported in Section 4. Conclusions are given in Section 5.

2. GENETICS-BASED KNOWLEDGE INTEGRATION

Here, we assume that all knowledge sources are represented by rules since almost all knowledge derived using knowledge-acquisition tools or induced using machine-learning methods may easily be translated into or represented by rules.

The proposed approach uses the genetic algorithm to maintain a population of initial rule sets and automatically searches for the best integrated rule set. It consists of two phases: *encoding* and *integration*. The encoding phase transforms each rule set into a bit-string structure. The integration phase chooses bit-string rule sets for “mating” and gradually creates good offspring rule sets. The offspring rule sets then undergo recursive “evolution” until an optimal or nearly optimal rule set is found. The proposed algorithm is presented below.

Knowledge Integration Algorithm:
Input: *m* rule sets from different knowledge sources and a set of test instances.
Output: one integrated rule set that performs well.

Knowledge Encoding Phase:

Step 1: Collect multiple rule sets from multiple experts or using various machine learning methods.

Step 2: Transform each rule set into an intermediary representation.

Step 3: Encode the intermediary representation as a bit string that will act as an individual in the initial population.

Knowledge Integrating Phase:

Step 4: Evaluate the fitness value of each rule set using an evaluation function and a set of test instances.

Step 5: Select “good” rule sets upon which to perform the following genetic operations:
a: *Dynamic crossover* on parent rule sets to generate *offspring* rule sets;
b: *Mutation* on parent rule sets to generate *offspring* rule sets;

Step 6: Evaluate the fitness value of each rule set using an evaluation function and a set of test instances.

Step 7: If the termination criterion (such as a given number of generations, a given processing time, or convergence of fitness values) has been reached, then GO TO STEP 8; otherwise, GO TO STEP 5.

Step 8: Select the best rule set from the population as the final knowledge base.

These two phases are described in detail in the following sections.

2.1 Knowledge Encoding

Since rule sets from different knowledge sources may differ in size and rule set sizes may not be known beforehand, we encode knowledge as classifier systems with genetic operations, and credit assignment is applied at the rule-set level do [4, 7]. Variable-length bit strings are then used to represent rule sets. We first construct an intermediary representation to retain the syntactic and semantic constraints of each classification rule. Each intermediary representation is composed of *N feature tests* and one *class pattern,* where *N* is the number of features. Each feature test is then encoded into a fixed-length binary string, the length of which is equal to the number of possible feature test values. Thus, each bit represents a possible value. Similarly, the class pattern is encoded into a fixed-length binary string with each bit representing a possible class.

Example 1: Assume that brain tumors are to be diagnosed; two classes {*Adenoma, Meningioma*} will be distinguished using three features {*Location, Calcification, Edema*}. Assume that Feature *Location* has three possible values {*brain surface, sellar, brain stem*}, that Feature *Calcification* has four possible values {*no, marginal, vascular-like, lumpy*}, and that Feature *Edema* has three possible values {*no,* < 2 *cm,* < 0.5 *hemisphere*}. Also assume that a rule set RS_i from a knowledge source has only the following two rules:

R_1 : If (*Location* = *sellar*) and (*Calcification* = *no*) then *Class* is *Adenoma*;
R_2 : If (*Location* = *brain surface*) and (*Edema* < 2 *cm*) then *Class* is *Meningioma.*

After translation, the intermediary representations of these rules are then be constructed as follows:

R'_1 : If (*Location* = *sellar*) and (*Calcification* = *no*) and ***(Edema = no or Edema < 2 cm or Edema < 0.5 hemisphere)***, then *Class* is *Adenoma*;

R'_2 : If (*Location* = *brain surface*) and ***(Calcification = no or Calcification = marginal or Calcification = vascular like or Calcification = lumpy)*** and (*Edema* < 2 *cm*) then *Class* is *Meningioma.*

The underlined tests are *dummy tests*. Also, R_1 and R_2 are logically equivalent to $R_1^{'}$ and $R_2^{'}$.

Using the intermediary form, we encode each feature test into a fixed-length binary string. For example, the set of legal values for feature *Location* is {*brain surface, sellar, brain stem*}; three bits are then used to represent this feature. The bit string 101 represents the test for *Location*, which is "*brain surface*" or "*brain stem*". As a result, the above rules are, respectively, encoded as follows:

	Location	*Calcification*	*Edema*	*Class*			*Location*	*Calcification*	*Edema*	*Class*
R_1'	010	1000	111	10		R_2'	100	1111	010	01

Finally, rule set RS_i is encoded into the string "01010001111010011110 1001".

2.2 Knowledge Integration

The proposed genetic knowledge-integration algorithm requires that a population of individuals must be initialized during the evolution process. In our approach, the initial set of bit strings for rule sets comes from the multiple knowledge sources. Each rule set represents one individual in the initial population.

In order to develop a "good" knowledge base from an initial population of rule sets, the accuracy and complexity of the resulting knowledge structure are used to evaluate the derived rule sets. Accuracy is evaluated using training instances as follows:

$$Accuracy(RS_i) = \frac{thetotal\,numberof\ testobjects correctly\ predicted\,by\,RS_i}{thetotal\,numberof\ training\ objects},$$

where RS_i is the i-th resulting rule set. The complexity of a resulting rule set (RS_i) is evaluated using the ratio of rule increase, defined as follows:

$$Complexity\,(RS_i) = \frac{Number\ of\ rules\,within\ the\ integrated\,rule\ set\ \ RS_i}{[\sum_{j=1}^{m}(Number\ of\ rules\,within\ initial\ \ RS_j)]\,/\,m},$$

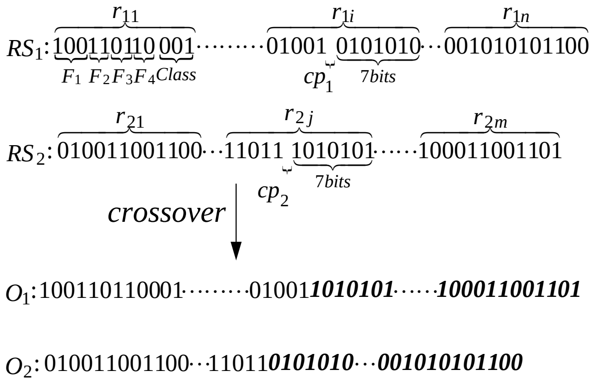
where RS_j is the j-th initial rule set and m is the number of initial rule sets. Accuracy and complexity are combined to represent the fitness value of the rule set. The evaluation function for a rule set RS_i is defined as follows:

$$Fitness(RS_i) = \frac{Accuracy(RS_i)}{[Complexity\,(RS_i)]^{\alpha}},$$

where α is a control parameter, representing a trade-off between accuracy and complexity. If the α value is small, the fitness function then focuses on the classification accuracy. On the contrary, if the α value is large, the fitness function is then dominated by the complexity.

During evolution, *dynamic crossover* and *mutation* operators are applied to the population of rule sets for knowledge integration. *Dynamic crossover* operators select crossover points differently from the way in which crossover operators are selected in the simple genetic algorithm. The original crossover operator chooses the same points for both parent chromosomes, but the dynamic crossover operator does not need to use the same point positions for both parent chromosomes. Dynamic crossover points may occur within rule strings or at rule boundaries. The only requirement for *dynamic crossover* points is that they “match up semantically”. That means that, if one parent is cut at a rule boundary, then the other parent must also be cut at a rule boundary. Similarly, if one parent is cut at a point *p* bits to the left of a rule boundary, then the other parent must also be cut at a point *p* bits to the left of some other rule boundary. The parents then generate offspring rule sets in search of the best integrated rule set. An example of a dynamic crossover operation is given below.

Example 2: Assume that two parent rule sets, RS_1 and RS_2, respectively, contain *n* and *m* rules with four features (F_1, F_2, F_3, and F_4). Feature F_1 has three possible values; features F_2, F_3, and F_4 all have two possible values. Three classes are to be determined. If crossover point cp_1 is the seventh bit to the left of r_{2_i} in RS_1, then crossover point cp_2 in RS_2 must be the seventh bit to the left of a certain rule r_{2j}. Thus, the crossover operator generates two offspring rule sets, O_1 and O_2, as shown in Fig. 1.



r11
r1i
r1n
RS1: 100110110 001 ········ 01001 0101010 ·· 001010101100
F1 F2 F3 F4 Class
cp1
7bits
r21
r2j
r2m
RS2: 010011001100 ·· 11011 1010101 ····· 100011001101
cp2
7bits
crossover
O1: 100110110001 ········ 01001**1010101** ····· **100011001101**
O2: 010011001100 ·· 11011**0101010** ·· **001010101100**

Fig. 1. An example of a crossover operation.

The mutation operator is the same as the standard one in the simple genetic algorithm. It randomly changes some elements in a selected rule set to help the integration process escape from local-optimum “traps”.

3. KNOWLEDGE REFINEMENT

A knowledge base consisting of multiple integrated knowledge sources is often only a prototype, with unsatisfactory classification accuracy. During the inference process, an input event wrongly classified by the current knowledge base causes a fault. The faulty rules in a knowledge base must be refined to improve the effectiveness of the knowledge base system [1]. In this section, the refinement scheme uses the knowledge-integration procedure as the basis for refining the knowledge.

The refinement scheme refines the knowledge base whenever the expert identifies a fault and provides a correct answer for the wrongly solved event. This event-solution pair is, thus, used as a training case for the refinement process to alter the knowledge base. It is, thus, appended to the training set for evaluation of the fitness function. Also, it is encoded as a bit string and appended to the current best rule set, thus enabling the search to starts at a good position. The new population size is the same as the one obtained using the knowledge-integration approach. The new training set including the wrongly classified event, is then presented to the refinement mechanism so that rule sets can be evaluated for a new population. The refinement process works until the exception event can be correctly classified by the knowledge base, making the new knowledge base more accurate than the old one. The proposed knowledge-refinement algorithm mentioned above is presented below.

Knowledge Refinement Algorithm:

Input: A current knowledge base, a current training set, and an input event wrongly classified by the current knowledge base.
Output: One refined rule set.

Step 1: Execute the knowledge-encoding phase and generate an initial knowledge population.
Step 2: Execute the knowledge-integration phase to generate the best rule set according to the current population.
Step 3: Execute the inference process according to the input events.
Step 4: Execute the knowledge-refinement phase whenever an input event wrongly classified by the current knowledge base causes a fault. The refinement process is made up of the following substeps:
a: Interpret a fault and provide the correct answer for the wrongly solved event from experts.
b: Encode the event-solution pair as a bit string and append it to the current knowledge base as a new individual in the population.
c: Add the event-solution pair to the current training set to form a new set.
d: Execute Step 2.

4. EXPERIMENTAL RESULTS

The brain tumor diagnostic problem [8, 9] was used to test the performance of the proposed two-phase genetic knowledge-integration approach. The 504 cases used in these experiments were obtained from Veterans General Hospital (VGH) in Taipei, Taiwan. Each case was expressed in terms of 12 features and a pathology. The goal was to identify one of six possible classes of brain tumors, including *Pituitary Adenoma*, *Meningioma*, *Medulloblastoma*, *Glioblastoma*, *Astrocytoma*, and *Anaplastic Protoplasmic Astrocytoma*, which are frequently found in Taiwan.

The 504 cases were first divided into two groups, a training set and a test set. The training set was used to evaluate the fitness values of rule sets during the integration and refinement processes; the test set provided as input events which could be used to test the resulting rule set, and the percentage of correct predictions was recorded. In each run, 70% of the brain tumor cases (353 cases) were selected at random for training, and the remaining 30% of the cases (151 cases) were used for testing. Ten initial rule sets were obtained from different groups of experts at VGH or derived using machine learning methods [2, 3, 6]. Each rule was encoded into a bit string 105 bits long. The accuracy of the ten initial rule sets was measured using the test instances. The results are shown in Table 1.

Table 1. The accuracy of the ten initial rule sets.

Rule Sets	Accuracy	No. of rules	Rule Sets	Accuracy	No. of rules
Rule Set 1	60.03%	52	Rule Set 6	77.89%	56
Rule Set 2	79.81%	56	Rule Set 7	68.53%	52
Rule Set 3	73.24%	56	Rule Set 8	72.83%	53
Rule Set 4	64.74%	53	Rule Set 9	76.24%	56
Rule Set 5	58.67%	52	Rule Set 10	70.19%	53

Although the ten initial rule sets were not accurate enough, they nevertheless could serve as a set of locally-optimal solutions that indicated useful information in the search space. Beginning with these rule sets, the proposed genetic knowledge-integration approach could then more rapidly reach the (nearly) optimal global solution than it could if it had nothing to refer to.

In the experiments, the *crossover* and *mutation* ratios were set at 0.9 and 0.04 respectively. Here, α was set at 0.125. The selection strategy used in both phases was the fitness-proportionate-selection strategy (FPS) [5]. The fitness proportionate selection strategy was used to select pairs of individuals in the population to generate new individuals. Among the new individuals and the original individuals in the population, those with high fitness values were passed to the new generation. The knowledge-integration algorithm achieved an accuracy rate of 84.76% after 2000 execution generations (11238.2 seconds). The size and the complexity of the resulting knowledge base were respectively, 86 and 1.595. Note that the accuracy rate was higher than that for any initial rule set shown in Table 1. Fig. 2 shows the relationship between the number of generations and the fitness value of the best rule set for the proposed approach.

As the number of generations increased, the resulting fitness value also increased, finally converging to about 83. Although the resulting rule set achieved an accuracy rate of 84.76%, 23 cases were nevertheless misclassified by this knowledge base. Thus, rules in the knowledge base must be refined to improve the effectiveness of the knowledge base. Experimental results, including accuracy, number of rules in the resulting rule set, and the refinement time, for different generations in the knowledge refinement are shown in Table 2.

The experimental results show that the knowledge refinement process can effectively improve accuracy although it requires some CPU time.

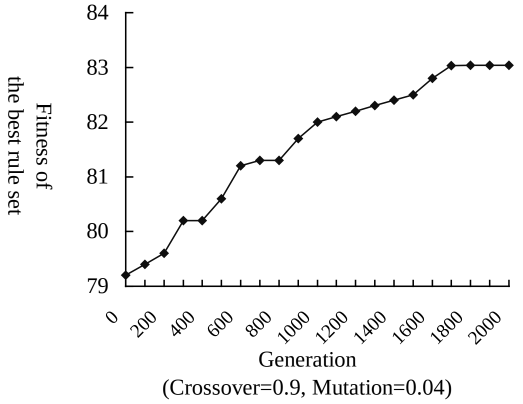


Fig. 2. Relationship between the fitness values of the best rule set and generations for the brain tumor domain.

Table 2. The experimental results for the knowledge refinement.

Rule Sets	Accuracy	No. of rules	CPU Time (second)
Initial refinement	84.76%	86	-
Refinement (10 generations)	89.23%	88	5.6
Refinement (50 generations)	96.01%	89	280.4
Refinement (100 generations)	97.32%	90	561.9

5. CONCLUSIONS AND DISCUSSION

In this paper, we have proposed a genetics-based knowledge-integration approach to effectively integrate multiple rule sets. The experimental results show that the rule set derived using our proposed approach has the following advantages over conventional knowledge-integration systems:

1. Only a small amount of computation time is needed compared to that required by human expert knowledge integration.
2. A large number of rule sets can be effectively integrated.
3. Domain experts need not intervene in the integration process.
4. It is objective since human experts are not involved in the integration process.

Furthermore, a knowledge refinement scheme based on the proposed knowledge-integration approach has been proposed rule refinement during the inference process. The experimental results show that the proposed refinement scheme can effectively improve the derived initial knowledge base. The proposed knowledge-integration approach and refinement scheme have been applied to the brain tumor domain and have yielded superior accuracy.

Although the work presented here shows good results, it is only a beginning. Much work still has remains to be done in this field.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their very constructive comments.

REFERENCES

1. T. R. Addis, "Knowledge refining for a diagnostic aid," *International Journal of Man-Machine Studies*, Vol. 17, 1982, pp. 151-164.
2. J. Cendrowska, "PRISM: An algorithm for inducing modular rules," *International Journal of Man-Machine Studies*, Vol. 27, 1987, pp. 349-370.
3. P. Clark and T. Niblett, "The CN2 induction algorithm," *Machine Learning*, Vol. 3, 1989, pp. 261-283.
4. K. A. DeJohn, "Learning with genetic algorithm: an overview," *Machine Learning*, Vol. 3, 1988, pp. 121-138.
5. J. H. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: University of Michigan Press, 1975.
6. J. Quinlan, "Induction of decision tree," *Machine Learning*, Vol. 1, 1986, pp. 81-106.
7. C. H. Wang, T. P. Hong, S. S. Tseng, and C. M. Liao, "Automatically integrating multiple rule sets in a distributed knowledge environment," *IEEE Transactions on Systems, Man, and Cybernetics-Part C*, Vol. 28, No. 3, 1998, pp. 471-476.
8. C. H. Wang, S. S. Tseng, and T. P. Hong, "Design of a self-adaptive brain tumor diagnostic system," *Journal of Information Science and Engineering*, Vol. 11, 1995, pp. 275-294.
9. C. H. Wang, T. P. Hong, and S. S. Tseng, "Self-integrating knowledge-based brain tumor diagnostic system," *Expert Systems With Applications*, Vol. 11, No. 3, 1996, pp. 351-360.
10. 10. C. H. Wang, T. P. Hong, and S. S. Tseng, "Knowledge integration by genetic algorithms," in *Proceedings of the Seventh International Fuzzy Systems Association World Congress*, 1997, pp. 404-408.
11. C. H. Wang, T. P. Hong, and S. S. Tseng, "Integration membership functions and fuzzy rule sets from multiple knowledge sources," *Fuzzy Sets and Systems*, 1998, to appear.
12. C. H. Wang, T. P. Hong, and S. S. Tseng, "A genetic fuzzy-knowledge integration framework," *IEEE International Conference on Fuzzy Systems*, Vol. 112, No. 1, 2000, pp. 141-154.
13. C. H. Wang, T. P. Hong, and S. S. Tseng, "Genetic-fuzzy knowledge-integration strategies," *10th IEEE International Conference on Tools With Artificial Intelligence*, 1998, pp. 250-255.

Ching-Hung Wang (王景弘) received the B.S. degree in computer and information science from Soochow University in 1984 and his Ph.D. degree in Computer and Information Science from National Chiao Tung University in 1997. Currently, he is an assistant researcher at Chunghwa Telecommunication Laboratories. His research interests are machine learning, genetic algorithms, neural networks, and fuzzy logic.

Tzung-Pei Hong (洪宗貝) received his B. S. degree in chemical engineering from National Taiwan University in 1985 and his Ph.D. degree in Computer Science and information engineering from National Chiao Tung University in 1992.

From 1987 to 1994, he was with the Laboratory of Knowledge Engineering, National Chiao-Tung University, where he was involved in applying techniques of parallel processing to artificial intelligence. From 1992 to 1994, he was an Associate Professor in the Department of Computer Science at the Chung-Hua Polytechnic Institute. He is currently an Associate Professor in the Department of Information Management at I-Shou University and an Associate Researcher with the National University of Kaohsiung in Preparation. His current research interests include parallel processing, machine learning, neural networks, fuzzy sets, expert systems, management information systems, and www applications.

Dr. Hong was the winner of the 1992 Acer Long Term Award for outstanding Ph.D. Dissertation. He is also a member of the Association for Computing Machinery, the IEEE Computer Society, the Chinese Fuzzy Systems Association and the Institute of Information and Computing Machinery.

Shian-Shyong Tseng (曾憲雄) received the Ph.D. degree in computer engineering from National Chiao Tung University in 1984. Since August, 1983, he has been on the faculty of the Department of Computer and Information Science at National Chiao Tung University, and is currently a Professor there. From 1988 to 1991, he was the Director of the Computer Center at National Chiao Tung University. From 1991 to 1992 and from 1996 to 1998, he acted as the Chairman of the Department of Computer and Information Science. From 1992 to 1996, he was the Director of the Computer Center at the Ministry of Education and the Chairman of Taiwan Academic Network (TANet) management committee. His current research interests include parallel processing, expert systems, computer algorithms, and Internet-based applications.

Dr. Tseng is an associate editor of Information and Education, and a member of the IEEE and Phi Tau Phi Societies. He was named an Outstanding Talent of Information Science of the Republic of China in 1989. He received the 1992, 1994, and 1995 Outstanding Research Awards from the National Science Council of the Republic of China. He was the winner of the 1990 and 1991 Acer Long Term Awards for outstanding M.S. Thesis Supervision and the winner of the 1992 and 1996 Acer Long Term Awards for outstanding Ph.D. Dissertation Supervision. He was also awarded the Outstanding Youth Honor of the R.O.C. in 1992.