73. Extended Genetic Programming Using Apriori Algorithm for Rule Discovery

Ayahiko Niimi and Eiichiro Tazaki

Department of Control and Systems Engineering,
Toin University of Yokohama
1614 Kurogane-cho, Aoba-ku, Yokohama 225-8502, JAPAN
tazaki@intlab.toin.ac.jp

Genetic programming (GP) usually has a wide search space and can use tree structure as its chromosome expression. So, GP may search for global optimum solution. But, in general, GP's learning speed is not so fast. Apriori algorithm is one of algorithms for generation of association rules. It can be applied to large database. But, It is difficult to define its parameters without experience. We propose a rule discovery technique from a database using GP combined with association rule algorithm. It takes rules generated by the association rule algorithm as initial individual of GP. The learning speed of GP is improved by the combined algorithm. To verify the effectiveness of the proposed method, we apply it to the meningoencephalitis diagnosis activity data in a hospital. We got domain expert's comments on our results. We discuss the result of proposed method with prior ones.

73.1 Introduction

Various techniques have been proposed for rule discovery using classification learning. In general, the learning speed of a system using genetic programming (GP) [73.1] is slow. However, a learning system which can acquire structural knowledge by adjusting to the environment can be constructed, because GP's chromosome expression is tree structure, and the structure is evaluated by fitness value for the environment.

On the other hand, there is the Apriori algorithm [73.2], a rule generating technique for large databases. This is an algorithm for generation of association rules. The Apriori algorithm uses two indices for rule construction: a support value and a confidence value. Depending on the setting of each index threshold, the search space can be reduced. However, it is possible that an unexpected rule cannot be extracted by reducing the range of the search space. Moreover, the load of the expert who analyzes the rule increases when there are a lot of association rule candidates, and it is a possible that it becomes difficult to search for a useful rule. Some experience is necessary to set an effective threshold.

Both techniques have advantages and disadvantages as above. In this paper, we propose an extended genetic programming using apriori algorithm for rule discovery. By using the combined rule generation learning method,

T. Terano et al. (Eds.): JSAI 2001 Workshops, LNAI 2253, pp. 525−532, 2001.

it is expected to construct a system which can search for high accurate rules in large databases. The purpose of this research is achieving high forecast accuracy by small number of rules.

73.2 Genetic Programming

Genetic programming (GP) is a learning method based on the natural theory of evolution, and the flow of the algorithm is similar to genetic algorithm (GA). The difference between GP and GA is that GP has extended its chromosome to allow structural expression using function nodes and terminal nodes. [73.1] In this paper, the tree structure is used to express the decision tree.

The decision tree construction by GP follows the following procedures.

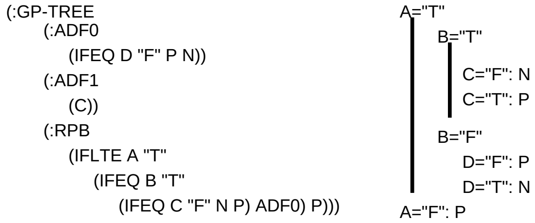
1. An initial population is generated from a random grammar of the function nodes and the terminal nodes defined for each problem domain.
2. The fitness value, which relates to the problem solving ability, for each individual of the GP population is calculated.
3. The next generation is generated by genetic operations.
   a) The individual is copied according to the fitness value (reproduction).
   b) A new individual is generated by intersection (crossover).
   c) A new individual is generated by random change (mutation).
4. If the termination condition is met, then the process ends. Otherwise, the process repeats from the calculation of fitness value in step 2.

Generally, there is no method of adequately controlling the growth of the tree, because GP does not evaluate the size of the tree. Therefore, during the search process the tree may become overly deep and complex, or may settle to a too simple tree. The technique by which GP defines an effective partial tree is proposed. The approache is automatic function definition (or Automatically Defined Function: ADF), and this is achieved by adding the gene expression for the function definition to normal GP [73.4]. By implementing ADF, a more compact program can be produced, and the number of generation cycles can be reduced. More than one gene expression of ADF can be defined in one individual.

One example of our GP expression is shown following.(See Figure 73.1)

In Figure 73.1, decision tree is expressed in the form similar to LISP-code. GP-TREE expresses one individual of GP, and GP-TREE is composed of the ADF definition part and the main tree part. "RPB" defines main GP tree. Both "ADF0" and "ADF1" defined as each ADF tree. "IFLTE", "IFEQ" are function nodes. These functions requires four arguments(in following example, we use $arg1, arg2, arg3, arg4$). The definitions of them are following.

(IFLTE $arg1, arg2, arg3, arg4$) if $arg1$ is less than or equal to ($\leq$) $arg2$ then evaluate $arg3$, else then evaluate $arg4$



```
(:GP-TREE
   (:ADF0
      (IFEQ D "F" P N))
   (:ADF1
      (C))
   (:RPB
      (IFLTE A "T"
         (IFEQ B "T"
            (IFEQ C "F" N P) ADF0) P)))
```

```
A="T"
|  B="T"
|  |  C="F": N
|  |  C="T": P
|  B="F"
|     D="F": P
|     D="T": N
A="F": P
```

Fig. 73.1. Expression of GP's Chromosome (The left side is an individual expression of LISP-code and the right side is rewritten to the decision tree expression.)

(IFEQ $arg1, arg2, arg3, arg4$) if $arg1$ is equal to($=$) $arg2$ then evaluate $arg3$, else then evaluate $arg4$.

A, B, C and D express the attributes in database. "T" and "F" express attribute value, and "N" and "P" express class name.

73.3 Approach of Proposed Combined Learning

To make up for the advantage and the disadvantages of the Apriori algorithm and GP, we propose a rule discovery technique which combines GP with the Apriori algorithm. By combining each technique, the search of high accurate rules from a large database is expected. An outline of our proposed technique is shown in Figure 73.2.

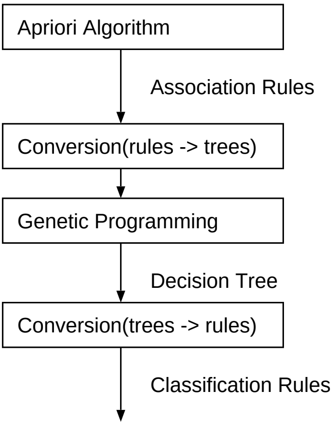


Fig. 73.2. Flow Chart of Approach of Proposed Combined Learning

The following steps are proposed for the rule discovery technique.

1. First, the Apriori algorithm generates the association rule.
2. Next, the generated association rules are converted into decision trees which are taken in as initial individuals of GP. The decision trees are trained by GP learning.
3. The final decision tree is converted into classification rules.

This allows effective schema to be contained in the initial individuals of GP. As a result, it is expected to improve the GP's learning speed and its classification accuracy. However, when GP is used for multi-value classification, the learning speed of GP may become slow due to increasing the number of definition nodes. Therefore, it is difficult to apply the proposed technique to multi-value classification.

For conversion from the association rule into decision trees, we use the following procedures.

1. For the first process, the route of the decision tree is constructed, assuming the conditions of the association rule as the attribute-based tests of the decision tree.
2. In the next process, the conclusions of the association rule is appended on the terminal node of this route.
3. Finally, the class value of the terminal nodes which are not defined by the association rule are assigned by randamly choosing from the terminal nodes set.

In conversion from the association rule to the decision tree, a rule which contains class attribute in the conclusion part is selected. One decision tree is converted based on one association rule. A too simple decision tree is generated by conversion, but the decision tree of high accuracy is not necessary to GP's initial individuals, because of GP learning. The conversion does not make the amount of the calculation increase because it is simple conversion. For conversion from the GP's decision tree to the classification rule, we use the process proposed by Quinlan [73.5].

73.4 Apply to Rule Discovery from Database

We applied the proposed technique for the meningoencephalitis diagnosis data sets. This database was donated by S.Tsumoto[73.6]. We applied the proposed technique for "find factors important for diagnosis (DIAG2) to judge bacteria or virus". We obtained following results of decision tree and rules generated by ADF-GP. In the proposed method, we took the association rule generated by Apriori algorithm as initial individuals of GP. We used 70 data for training, 140 data for test. 70 data was extracted at random. We studied these data by using the normal GP, and tuned of the GP parameter before experiment.

We defined some expressions. “A eq B” is express that attribute(A) is equal to attribute(B) if its attribute is discrete value. “A && B” represents to connect each part(A,B) of rules by ‘’and”. The left side of “→” express conditions of rule, and the right side of “→” express conclusion of rule (or class name).

The section 73.4.1 shows the results using ADF-GP only. The section 73.4.2 shows the results using proposed technique.

73.4.1 ADF-GP Only

The following rules are generated with ADF-GP. The generated rules are composed by the categorical attributes.

```
=== generated rules ===
rule1:
 (EEG_FOCUS eq "-") && (CT_FIND eq "normal") && (SEX eq "M")
&& (RISK eq "n") -> VIRUS
rule2:
 (EEG_FOCUS eq "-") && (CT_FIND eq "normal") && (SEX eq "M")
&& (RISK eq "p") -> BACTERIA
rule3:
 (EEG_FOCUS eq "-") && (CT_FIND eq "normal") && (SEX eq "F")
&& (RISK eq "n") -> VIRUS
rule4:
 (EEG_FOCUS eq "-") && (CT_FIND eq "normal") && (SEX eq "F")
&& (RISK eq "p") -> BACTERIA
rule5:
 (EEG_FOCUS eq "+") && (CT_FIND eq "abnormal") && (SEX eq "F")
&& (RISK eq "n") -> VIRUS
rule6:
 (EEG_FOCUS eq "+") && (CT_FIND eq "abnormal") && (SEX eq "F")
&& (RISK eq "p") -> BACTERIA
rule7:
 (EEG_FOCUS eq "+") && (CT_FIND eq "abnormal") && (SEX eq "M")
-> BACTERIA
rule8:
 (EEG_FOCUS eq "-") && (CT_FIND eq "abnormal") -> BACTERIA
rule9:
 (EEG_FOCUS eq "+") && (CT_FIND eq "normal") -> VIRUS
```

To examine the availability and the accuracy of the generated rule, the size of the rule, the use frequency and the wrong classification frequency (wrong classification rate) to all data, the classification class by rules are shown in Table(73.1). In the table, the rule 6 is not used for all data. The rules (1 and 3) with high availability show low wrong classification rates. Other rules have high wrong classification rate independent of availability.

To examine the classification accuracy of the generated rule set, each classification distribution to all data are shown in Table(73.2). The table shows that small number of data could not classify VIRUS and BACTERIA correctly.

73.4.2 Proposed Technique (Association Rules + ADF-GP)

The following rules are generated with proposed technique. The generated rules are composed by the continuous value attributes.

Table 73.1. Evaluation on test data by each rules (ADF-GP only)

Rule	Size	Used	Wrong		
1	4	33	4	(12.12 %)	: VIRUS.
2	4	6	2	(33.33 %)	: BACTERIA.
3	4	36	3	(8.33 %)	: VIRUS.
4	4	2	0	(0.00 %)	: BACTERIA.
5	4	7	0	(0.00 %)	: VIRUS.
6	4	0	0	(0.00 %)	: BACTERIA.
7	3	5	1	(20.00 %)	: BACTERIA.
8	2	27	8	(29.63 %)	: BACTERIA.
9	2	24	6	(25.00 %)	: VIRUS.

Table 73.2. Evaluation on test data by error distribution(ADF-GP only)

(a)	(b)	← classified as
87	11	(a):class VIRUS
13	29	(b):class BACTERIA

total hits= 116

```
=== generated rules ===
rule1:
 (Cell_Poly <= 221) -> VIRUS
rule2:
 (Cell_Poly > 221) && (EEG_FOCUS <= 200) -> BECTERIA
rule3:
 (Cell_Poly > 221) && (EEG_FOCUS > 200) && (GCS <= 121)
-> BECTERIA
rule4:
 (Cell_Poly > 221) && (EEG_FOCUS > 200) && (GCS > 121)
&& (SEIZURE == 0 ) -> VIRUS
rule5:
 (Cell_Poly > 221) && (EEG_FOCUS > 200) && (GCS > 121)
&& (SEIZURE != 0 ) -> BACTERIA
```

The performance of the generated rule are shown in Table(73.3). In the table, the rule 3, 4 and 5 are not used for all data. The rule 1 and 2 have high availability and low wrong classification rates.

Table 73.3. Evaluation on test data by each rules (proposed method)

Rule	Size	Used	Wrong		
1	1	108	10	(9.26 %)	: VIRUS.
2	2	32	0	(0.00 %)	: BACTERIA.
3	3	0	0	(0.00 %)	: BACTERIA.
4	4	0	0	(0.00 %)	: VIRUS.
5	4	0	0	(0.00 %)	: BACTERIA.

To examine the classification accuracy of the generated rule set, each classification distribution to all data is shown in Table(73.4). The table shows that some rules classified BACTERIA as VIRUS by mistake, but almost rules have correct classification ability.

Table 73.4. Evaluation on test data by error distribution (proposed method)

(a)	(b)	← classified as
98	0	(a):class VIRUS
10	32	(b):class BACTERIA

total hits= 130

73.4.3 Discussion for the Results

In the results, the proposed method shows higher accuracy than ADF-GP, and dataset can be expressed using more small number of rules.

The proposed method does not have pruning rules operation except for GP operations. GP operation is a kind of statistical operation. Thus, sometimes GP operation can obtain interesting rules, but otherwise, the result contains meaningless rules. For such problems, GP technique which contain the pruning operation are proposed [73.7], and it makes possible to build the pruning techniques in our proposed technique. Moreover, it is also possible in the experiment to remove meaningless rules by using the threshold in availability. When the experimental result is evaluated and cleaned by domain expert after experiment, the load for domain expert depends on the number of rules of results. In the proposed technique, the number of rules of results can be reduced compared with only ADF-GP.

We got following comments on these results from domain expert(S. Tsumoto).

> Totally, the results obtained by ADF-GP are more interesting than the proposed methods. The results obtained by the proposed technique are very reasonable, but I do not see the meaning of "EEG_FOCUS > 200" and "GCS > 121". Please let me know what the authors mean by that. Please show me the results for other problems.

The purpose of this research is to achieve high forecast accuracy by small number of rules. This purpose is not as same as expert's interest on the experiment result. Because expert's interesting rules were obtained by the normal ADF-GP, expert's interesting rule can be obtained by the proposed technique by increasing the GP effect.

73.5 Conclusions

In this paper, we proposed the rule discovery technique from the database using genetic programming combined with association rule algorithms. To verify the validity of the proposed method, we applied it to the meningoencephalitis diagnosis activity data in a hospital, and discussed the results of proposed method and normal ADF-GP with domain expert. As a result, an improvement of rules' accuracy was seen, and proposed method can express dataset by the small number of rules. It can be concluded that the proposed method is an effective method to the improvement of the rules' accuracy and can save the number of rules for the rule discovery problem. Though the comments of domain expert, using only ADF-GP method can be obtained more interesting rules than using proposed method.

In the future, we will research the following 4 topics. The first topic is to apply the method to other verifications. We already applied proposed method for other problems [73.8] [73.9]. We need to discuss the problem suitable for proposed method through the applications to various problems. The second topic is to discuss the conversion algorithm from the association rule to a decision tree with high accuracy. The third topic is to extend the proposed method to multi-value classification problems. It is necessary for this problem to suppress increasing the number of definition nodes and to establish measures against the decrease at the learning speed by increasing nodes. The fourth topic is to obtain more interesting rules such as ADF-GP only.

References

73.1 Koza, J. R. (1992): Genetic Programming. MIT Press.

73.2 Terabe, M., Katai, O., Sawaragi, T., Washio, T., Motoda, H. (2000): Attribute Generation Based on Association Rules. Journal of Japanese Society for Artificial Intelligence, Vol.15 No.1,pp.187–197 (Japanese).

73.3 Kitsuregawa, M. (1997): Mining Algorithms for Association Rules. Journal of Japanese Society for Artificial Intelligence, Vol.12 No.4,pp.513–520 (Japanese).

73.4 Koza, J. R., Kinner, K. E.(ed.), et.al (1994): Scalable Learning in Genetic Programming Using Automatic Function Definition. Advances in Genetic Programming, pp.99–117.

73.5 Quinlan, J. R. (1993): C4.5: Programs for Machine Learning. Morgan Kaufman Publishers.

73.6 Tsumoto, S. (2000): Meningoencephalitis Diagnosis data description. [http://www.ar.sanken.osaka-u.ac.jp/jkdd01/menin.htm].

73.7 Niimi, A., Tazaki, E. (1999): Extended Genetic Programming using Reinforcement Learning Operation. Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, pp.596–600.

73.8 Niimi, A., Tazaki, E. (2000): Genetic Programming Combined with Association Rule Algorithm for Decision Tree Construction. Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering System & Allied Technologies, volume 2,pp.746–749.

73.9 Niimi, A., Tazaki, E. (2000): Rule Discovery Technique Using Genetic Programming Combined with Apriori Algorithm. Proceedings of the Third International Conference on Discovery Science, pp.273–277.

73.10 Niimi, A., Tazaki, E. (2000): Knowledge Discovery using Extended Genetic Programming from Biochemical Data. Proceedings of 49th KBS meeting, pp.45–49, Japan AI Society, SIG-KBS-A002 (Japanese).