Extension of Hierarchically Classifier System using Self GA Invocation

Satoshi ENDOH† Hiroshi KAWAKAMI‡ Azuma OHUCHI‡

†Department of Information Engineering,Faculty of Engineering
University of the Ryukyus
‡Laboratory of Harmonious Systems Eng., Research Group of Complex Systems Eng.
Division of Systems and Information Eng., Faculty of Eng.
Hokkaido University

1 Introduction

Classifier system is one of the powerful and robust learning system which is able to learn good rule set in the nonlinear environments. Although classifier system shows great promise, the system has a number of serious difficulties about it's learning efficiency. This difficulty is reflected by the problem of rule clustering and rule association.

First problem, the rule clustering problem is that whole classifiers in a population will become similar patterns, but these patterns don't include whole sub-solutions. Second problem is rule association problem. Traditional classifier systems do not have effective mechanisms for creating and maintaining default hierarchies or rule chains.

Then we propose the multiple leveled hierarchically structured classifier system called mHCS. Although the mHCS system leaves many features of traditional CS, one of the important difference is that some classifiers are grouped together as a family in mHCS. Then mHCS has n-population levels that are classifier level, family level, meta family level,...,and n-meta family level. Members of a family work to maximize their family's strength and the each member's strength. In this system, matching operator is executed at the classifier level. However, the matched classifier's bid is calculated not only by the strength of classifier and its specificity, but also the strength of the family that it's belongs to. Most genetic operations, except to the mutation and some parts of crossovers, are executed at the family level. It means that the basic unit for the genetic operation is changed to the family from the classifier.

The main causes of the rule clustering problem are the competitions among good classifiers or their schemata, and the disruption of useful patterns by genetic operations. In traditional classifier system, crossover between two classifiers generate offsprings that are not compatible with the solution set. The mHCS solve this problem by imposing some relations over these classifiers. These related classifiers may stay in the same family with high probability of generating incompatible offsprings. And then, classifiers in mHCS are grouped into families by selecting competent relations, that could be default hierarchies or valid chains. Most genetic operations are doing between families. Therefore, the probability of breaking default hierarchies or classifiers' chains by genetic operations is reduced.

Another approach to get a good performance of CS is to control the timing of GA operator invocation. Our system is able to control the timing of GA-invocation monitoring the saturation of family's strength. Some experiments are designed to investigate the performance of the mHCS. The results of experiments represents that the mHCS shows good performance for the problems that have large search space.

2 Classifier system

The research field of Genetic Based Machine Learning(GBML) are divide broadly into two categories. One is the Michigan approach originally proposed by Holland, and the other is the

0-7803-3280-6/96/$5.00

Pittsburgh approach by Smith. CS-1 system developed by Holland is now called Classifier System, and this system is one of the most popular learning system. A classifier system(CS) consists of the three subsystems as follows.

1. Rule & Message System
2. Apportionment of Credit System
3. Genetic Algorithm

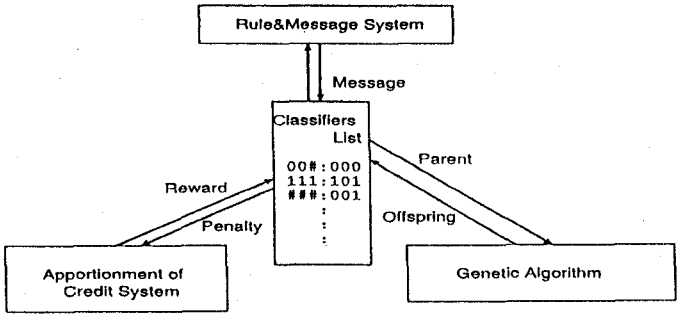

Figure 1: Classifier system

3 Problems

3.1 Rule clustering

First problem, the rule clustering problem is those whole classifiers in a population may overconcentrate to similar patterns, but these patterns don't include whole sub-solutions. The main causes of this problem are the competitions among good classifiers or their schemata, and the disruption of useful patterns by genetic operators. In traditional classifier systems, crossover between two classifiers may generate offsprings that are not compatible with the solution set. mHCS solve this problem by imposing some relations over these classifiers. These related classifiers may stay in the same family with high probability. If genetic operations are restricted in different families, HCS can reduce the probability of generating incompatible offsprings.

Another reason for the clustering problem is premature convergence. If a solution classifier is included in the initial population and more useful classifiers be generate much later, the first classifier would have a much higher strength in the population. Then the population may converge to this classifier prematurely. In HCS, the probability that at least a correct family is contained in an initial population is much smaller than at least one solution classifier is included in the initial population in a traditional classifier system. Therefore, the probability of converging to one family is smaller.

3.2 Rule association

Second problem is the rule association problem. Traditional classifier systems do not have effective mechanisms for forming and maintaining default hierarchies or rule chains. In mHCS, classifiers are grouped into families by selecting competent relations, that could be default hierarchies or valid chains. Most genetic operations are doing between families. Therefore, the probability of breaking default hierarchies or classifiers' chains by genetic operations is reduced.

3.3 Rule set association

mHCS is able to reduce the rule association problem, moreover the system is able to solve the rule-set association problem. In mHCS the families at HCS are grouped into meta-families by some relations, and the mHCS can form and maintain the classifier chain sets or default hierarchy sets. Therefore, the mHCS has more performance of correspondence to large search and solution spaces than the ordinary CS.

4 mHCS

CS uses no hierarchy in the population to search solutions. For large search and solution spaces problem, the performance of CS with multiple level that is called mHCS is better than ordinary CS. mHCS with two level hierarchy can reduce the rule association problem by joining some classifiers together as a family, but cannot reduce the rule-set association problem. With two population levels, mHCS can take care of default hierarchies and classifier chains, but is not able to take care of default hierarchies sets or classifier chains sets.

mHCS with multi level hierarchy can reduce the rule-set association problem by joining some families together as a meta-family, by joining some meta-families together as a meta-meta-family. With multiple population levels, mHCS can take care of classifier chains sets.

4.1 matching

As in traditional classifier systems, matching is done at the classifier level. However, the bid of a matched classifier is determined not only by the strength of the classifier and its specificity, but also the sum of the strength of the family that it belongs to and strength of meta-families that the family belongs to.

The strength of a family is the sum of the strength of whole the classifiers that contained in the family, and the strength of a meta-family(meta-meta-family) is the sum of the strength of whole the families(meta-families) that contained in the meta-family(meta-meta-family).

A matched classifier's bid is calculated by the following formula:

$$bid = \frac{A \times s_c \times s_f \times \sum s_m \times sp}{2^{l-sp}} \qquad (1)$$

where A is a constant, s_c is the strength of the classifier, s_f is the strength of the family that containing the classifier, s_m is the strength of the meta-families that containing the family and sp is the specificity of the classifier.

4.2 Crossover

In mHCS, there are three kinds of crossover. The crossover operations are usually done at the family level. The possibilities of crossing over at the classifier level and crossing over at the meta-family levels are smaller than crossing over at the family level.

4.2.1 crossover between classifiers

The first one is general crossover that crossing over two classifiers selected from different families.

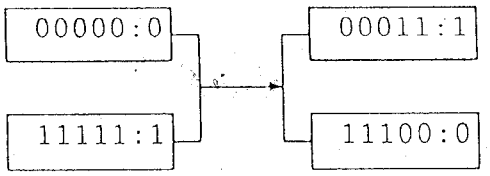

Figure 2: crossover between classifiers

4.2.2 crossover between families

The second one is crossing over two families selected from different meta-families. This kind of crossover has two situations.

situation 1. Swapping family members of the two selected families. In Figure 3, family members of the two selected families are swapped.

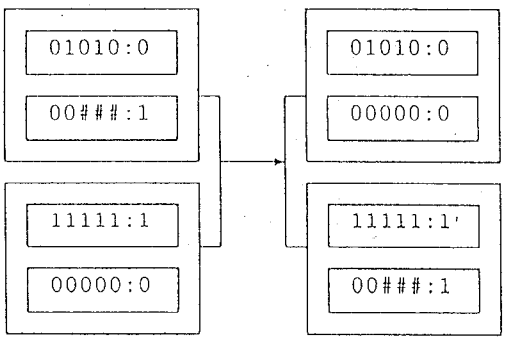

Figure 3: crossover between families

situation 2. Swapping part of each family member with that of the corresponding members in the other family.

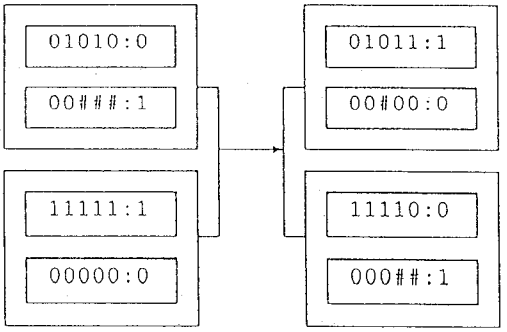

Figure 4: crossover between families

Figure 4 shows an example of the situation 2.

Consider a family as a matrix of classifiers, the above two situations' crossover can be think as swapping rows and swapping columns of the matrices.

In mHCS, classifiers are tied to a family, and probability of crossing between families is higher than other kinds. Therefore families are become the main part of genetic operations. Classifiers that belong to a family are related to each other by some relations. Then the competitions among these classifiers are restricted.

4.2.3 crossover between meta-families

The third one is crossing over two meta-families(or meta-meta-families). Figure 5 shows swapping families(meta-families) of the two selected meta-families(meta-meta-families).

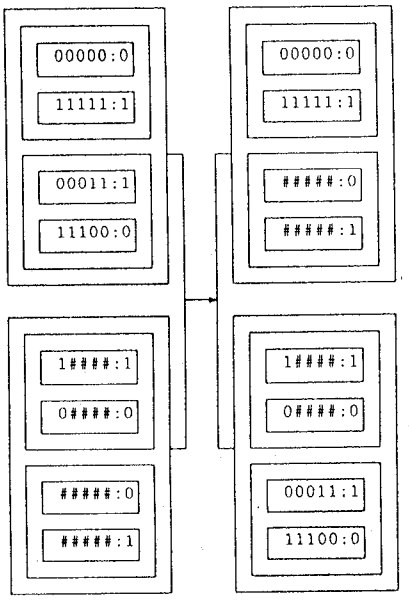

Figure 5: crossover between meta-families

Crossing over between meta-families is swapping families or meta-families. In other words, that is swapping default hierarchies set and classifier chains set. Therefore, by using this kind of crossover, mHCS can reduce the rule-set association problem. The mechanism of crossing over between meta-families can form and maintain set of default hierarchies and set of classifier chains. With this mechanism, mHCS is more useful than ordinary HCS, in large solution and search spaces.

4.3 mutation

In mHCS, mutation is done at the classifier level, as in traditional classifier systems. The operation changes one position on a classifier that selected randomly.

4.4 Experiments

4.4.1 Implementation of mHCS

mHCS is implemented as follows :

- $E = \{r_i\}$, where e_i $(i = 1, 2, ..., 2^l)$ is the binary string representation of a number between 0 and $2^l - 1$, l being the length of the strings. An environment state.
- $F = \{f_i\}$, where f_i $(i = 1, 2, ..., m)$ is a family.
- $MF = \{m_i f_j\}$, where $m_i f_j$ $(i = 1, 2, ...n, j = 1, 2, ..., m')$ is a meta-family. $m_2 f_j$ means a meta-meta-family, meta(2)family.
- Category G = $\{0, 1\}$.

A classifier contains two parts, the condition part and the action part. Both parts are formed from the alphabets {'0','1','#'}.

The relation over the classifiers in a family(over the families in a meta-family) is not default hierarchies(set) or valid chains(set). They are grouped by randomly.

The genetic operation used in this experiment is crossover, mutation is not used here. A matched classifier's bid is calculated by equation (1).

A general apportionment of credit algorithm(the bucket brigade algorithm) and the roulette wheel selection are used. The selection was used to select candidates for crossing over. Families (or classifiers in the case of crossing over at the classifier level or meta-families) with higher strength get a greater chance to be selected.

4.4.2 Problems

mHCS and CS were examined by learning Boolean functions. Several Boolean functions were used in the experiments to contrast the performance of mHCS and the performance of CS. The details of two examples of Boolean functions are as follows.

$$
\begin{aligned}
&f_1(x_0, x_1, x_2, x_3, x_4, x_5) = \\
&\quad (x_0 \; and \; x_1 \; and \; x_2) or (x_3 \; and \; x_4 \; and \; x_5) or \\
&\quad (x_1 \; and \; x_2 \; and \; x_4) or (x_0 \; and \; x_3 \; and \; x_5) or \\
&\quad (x_0 \; and \; x_2 \; and \; x_4) or (x_0 \; and \; x_1 \; and \; x_4) or \\
&\quad (x_1 \; and \; x_2 \; and \; x_3) or (x_2 \; and \; x_4 \; and \; x_5) or \\
&\quad (x_2 \; and \; x_3 \; and \; x_5) or (x_2 \; and \; x_4 \; and \; x_4) or \\
&\quad (x_0 \; and \; x_4 \; and \; x_5) or (x_0 \; and \; x_3 \; and \; x_4) or \\
&\quad (x_0 \; and \; x_1 \; and \; x_5).
\end{aligned}
$$

Figure 6: probrem1

Classifier is corded as follows

$$x_0 x_1 x_2 x_3 x_4 x_5 : x_6$$

where: The bit $x_6$is output bit and is compared with the value of f_1.

$$
\begin{aligned}
&f_2(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) = \\
&\quad (x_0\ and\ x_1\ and\ x_2) or (x_3\ and\ x_4\ and\ x_5) or \\
&\quad (x_1\ and\ x_2\ and\ x_4) or (x_0\ and\ x_3\ and\ x_5) or \\
&\quad (x_0\ and\ x_2\ and\ x_4) or (x_0\ and\ x_1\ and\ x_4) or \\
&\quad (x_1\ and\ x_2\ and\ x_3) or (x_2\ and\ x_4\ and\ x_5) or \\
&\quad (x_2\ and\ x_3\ and\ x_5) or (x_2\ and\ x_4\ and\ x_4) or \\
&\quad (x_0\ and\ x_4\ and\ x_5) or (x_0\ and\ x_3\ and\ x_4) or \\
&\quad (x_0\ and\ x_1\ and\ x_5) and X.
\end{aligned}
$$

Figure 7: probrem2

X is $10\times$ the value of addressx_6, x_7.
If $x_6 = 1, x_7 = 0$ then $X = x_2 \times 10$.
Classifier is corded as follows

$$x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 : x_8 x_9$$

The bits $x_8 x_9$ are output bits and is comparedf_2 the value of f_2.

The performances of these systems are determine as follows :

$$P = \frac{\sum_{t=1}^{T} (C_t - I_t)}{T} \qquad (2)$$

where C_t is the number of the correct answers at iteration t, I_t is the number of the incorrect answers at iteration t and T is the current iteration.

4.4.3 Results and discussions

The fig.8 shows performance curves for three HCS experiments and two CS experiments for problem1. The differences of HCS-4, HCS-8, and HCS-14 are setting of the parameters about family size and timing of GA invocation. And the difference of CS-3 and CS-0 is that the system uses crowding method or not. From the comparison of performance curves, mHCS shows the efficiency equal to or better than CS. But, the HCS curves shows a conspicuous vibration compared with CS curves. This feature means that the mHCS system is more sensitive than CS to the timing of GA invocation. Therefore, we have to propose new method to invocate genetic algorithm on suitable timing.

5 Self GA invocation

5.1 Problems of the mHCS

From the results of the above experiments, the mHCS has following merits.

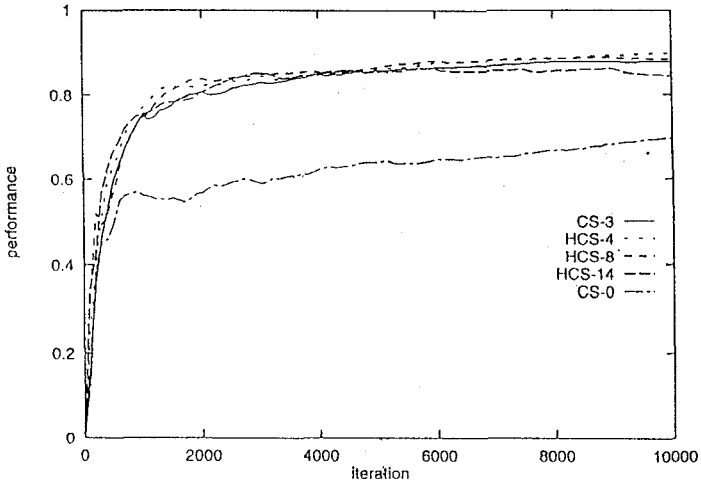

Figure 8: Performance curves

- reducing the rule clustering problem
- reducing the rule association problem

The other side, it becomes clear that the mHCS has a demerit of computational efficiency against the traditional CS. In the mHCS, we shifted the objects of genetic operations from classifiers to families. For this extension , the system becomes more sensitive to execution of genetic operations before classifier evaluation value is not fixed. In mHCS, if the timing of genetic operation execution is overhastily, there will be a strong probability that the system disrupts a lot of good classifiers. However, it's obvious that late execution of GA makes bad influences for computational efficiency. Then, we have to develop the system which is able to invocate GA on appropriate timing.

5.2 Strategy of Self GA invocation

In learning system like CS, genetic algorithms are used to discover new rules and genetic operations are applied to classifiers based on its evaluation value as a fitness value. Then, for GA works effectively as a rule discovering system, the evaluation values of each classifiers should be decided correctly. However, traditional CS adopts periodical GA invocation strategy, therefore useless crossover operator will be carried out so often. So we propose new strategy about the timing of GA invocation.

In the classifier system, if evaluation value of

each classifiers are decided by system, the system performance becomes saturation. Using this property, we propose a GA invocation rule as follows;

$$if(\sum_{k=t-2m}^{t-m} P(k) \cong \sum_{l=t-m}^{t} P(l))\ then\ (GAinvocation) \quad (3)$$

Where:

P(k)	:	system performance on timing k
t	:	current iteration number
m	:	constant.

This production rule shows that if there is no improvement about the system performance during m-th iteration, then the system invocates GA.

Using this strategy, the timing of GA invocation is just after the decision of the evaluation values for each classifiers. Accordingly, it is expected that

(1) reducing the invalid crossover operations,

(2) reducing the unnecessary evaluation of classifiers.

For this reasons, it is possible to improve the efficiency of mHCS. Next section, we show the effectivity of this strategy through the computational experiments.

5.3 Experiments

We tested the effects of the self-GA invocation method using previous problems.

This figure shows three performance curves of self-GA invocation method and one performance curve of CS. The difference of SI1, SI2 and SI3 is the frequency of GA invocation. This result shows that the efficiency of self-GA invocation method equal to or better than normal CS. From this results, the system becomes more effective and robust about control of GA. We are able to summarize the characteristics of mHCS with self-GA invocation as follows,

(1) The system is able to control useless GA incocation when the evaluation values of classifiers are not determined.

(2) The system is able to increase the number of invocation times when the evaluation values of populations are saturated.

In addition, we are able to reduce the labor of determine the parameter about GA invocation.

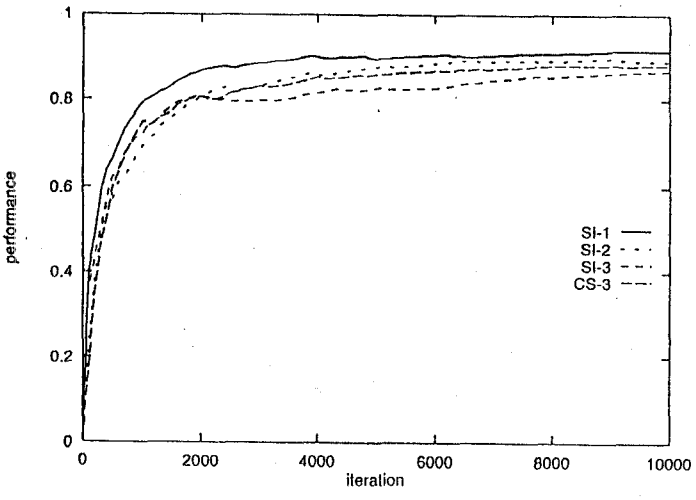

Figure 9: Performance curves of self-GA invocation

6 Conclusion

In classifier system techniques, an effective mechanism for forming and maintaining rule-set association is proposed. And the improvement of efficiency are discussed. The experiments showed effectiveness of mHCS and self-GA invocation method.

References

[Kawakami 95] *H,Kawakami, Satoshi ENDO and Azuma OHUCHI, "Extension of the HCS: multiple HCS" , Proc. of SICE'95,p.p.1615-1618, 1995*

[Goldberg 89] *Goldberg,D.E., "Genetic Algorithms in Search,Optimization and Machine Learning" , Addison-Wesley, 1989*

[Lingyan 91] *Lingyan Shu and Jonathan Schaeffer, "HCS: Hierarchies to Classifier Systems", Proc. of the 4th international conference on GA,1991*

[Lawrence 90] *Lawrence Davice, "Handbook of Genetic Algorithms", Thomson International Publishing,1990*