Co-evolutionary Rule-Chaining Genetic Programming

Wing-Ho Shum[1], Kwong-Sak Leung[1], and Man-Leung Wong[2]

[1] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong.
{whshum, ksleung}@cse.cuhk.edu.hk
[2] Department of Information Systems, Lingnan University, Tuen Mun, Hong Kong.
mlwong@ln.edu.hk

Abstract. A novel Genetic Programming (GP) paradigm called Co-evolutionary Rule-Chaining Genetic Programming (CRGP) has been proposed to learn the relationships among attributes represented by a set of classification rules for multi-class problems. It employs backward chaining inference to carry out classification based on the acquired acyclic rule set. Its main advantages are: 1) it can handle more than one class at a time; 2) it avoids cyclic result; 3) unlike Bayesian Network (BN), the CRGP can handle input attributes with continuous values directly; and 4) with the flexibility of GP, CRGP can learn complex relationship. We have demonstrated its better performance on one synthetic and one real-life medical data sets.

1 Introduction

GP is a branch of evolutionary computation (EC). It has been applied on different areas, like shortest path finding and classification[5]. Wilson's XCS [12] and XCSI [13] are well-known classification algorithms based on learning classifier system. However, the classification problems currently being addressed are single class problem.

BN is a network model, which represents a set of attributes for a given multi-class problem, and provides the probabilistic relationship among them. However, BN cannot handle continuous values directly; and the continuous values must be discretized first [6]. Heckerman et al. proposed methods of learning a network that contains Gaussian distributions [7]. Monti et al. use neural networks to represent the conditional densities [10].

In this paper, we propose the CRGP to handle the multi-class problem. It learns a set of classification rules, which represent the relationships among the attributes. It avoids cyclic rules; and it can handle input attributes with continuous values directly. The remaining parts of the paper are organized as follows. We describe the proposed algorithm in the next section. The experimental results are presented in Section 3. The summary appears in the last section.

2 Co-evolutionary Chaining Genetic Programming

The problem addressed in this study is to learn classification rules and the relationships among attributes without cycle in the inference process for multi-class problem, in which one or more than one attributes are regarded as classes. Each attribute of the problem is regarded as either an input attribute or a class. The relationships among the attributes will be represented by rules.

The CRGP is a novel coarse grained multi-population GP based on backward chaining. Multi-population maintain the diversity of rules. Backward chaining can perform classification through the inference process and cyclic rules have to be avoided.

2.1 Initialization of Populations

The CRGP uses the Michigan approach. It starts with n populations of rules, where n is equivalent to the number of classes. Each population is assigned to learn a different class, which is considered as the local class of the corresponding population. The other classes are referred to as foreign classes.

The rules represent the relationships among attributes. They are of the form, $< antecedent > \rightarrow < consequent >$. The rules are in the prefix form. The function set is defined as $F = \{\wedge, >, <=, =, \neq\}$.

Each population has a different copy of the data set, which contains the values of the input attributes and local class only. The foreign classes' values are omitted, i.e. the populations cannot access their foreign classes' values.

The training process is divided into a number of epochs, which in turn consists of a number of generations. The number of rules in each population is the same and static. The populations are initialized randomly.

2.2 Backward Chaining

The populations in the CRGP cooperate through rule migration. Migration occurs in the beginning of each epoch. During migration, the populations send a copy of their rules to the others. In other words, the populations would get a set of rules from the rest and they are referred to as migrated rules. To make it clear, hereafter, we refer to the populations' own rules as local rules. The new migrated rules always replace the existing migrated ones of previous epoch.

Backward chaining is a well-known inferential methodology. Given a class, some facts and rules, it forms a backward chain of rules and proves if the class can be satisfied. The CRGP employs backward chaining in the fitness evaluation and the cyclic relationship in the rules will be detected and eliminated.

Tables 1 show the pseudocodes of the fitness evaluation and backward chaining procedures respectively. The populations cannot access the foreign classes' values. During the fitness evaluation, if the local rule being evaluated contains foreign class in its antecedent part, the backward chaining procedure would be invoked, for each of these foreign classes.

Table 1. (Left) The pseudocode of the Fitness Evaluation Procedure. (Right) The pseudocode of the Backward Chaining Procedure.

1. Set $i = 0$.
2. while $i <$ the number of local rules,
   - if the i^{th} local rule contains one or more foreign classes in its $< antecedent >$,
     - ◇ the i^{th} local rule becomes the first rule in the backward chain of rules.
     - ◇ the backward chaining procedure is invoked for each of these foreign classes.
   - evaluate the i^{th} rule's fitness by the fitness function.

input: a foreign class being looking for
a backward chain of rules

1. Set $j = 0$ and $k = 0$.
2. while $j <$ the number of migrated rules,
   - if the j^{th} migrated rule infers the values of the foreign class, which is being looking for,
     - ◇ if the j^{th} migrated rule form no cycle with the others in the backward chain of rules
       - ○ the j^{th} migrated rule is selected.
3. Sort the selected migrated rules, according to their fitness values.
4. while $k <$ the number of selected migrated rules,
   - if the k^{th} selected migrated rule contains one or more foreign classes in its $< antecedent >$,
     - ◇ the k^{th} selected migrated rule is appended to the backward chain of rules.
     - ◇ another copy of backward chaining procedure is invoked for each of these foreign classes
   - fire the k^{th} selected migrated rule.
5. remove the last rule from the backward chain of rules.

The backward chaining procedure selects suitable migrated rules to infer the foreign class's values. The migrated rules with the following characteristics are selected: 1) it is a migrated rule inferring that foreign class's values; and 2) it will not form cycles with the others in the backward chain of the relevant rules.

The backward chaining procedure then fires the selected migrated rules one by one to infer the foreign class's values. The firing order is based on their fitness values, the one with the highest fitness value is fired first. The fitness values are brought from their original populations during migration. If a selected migrated rule does further contain foreign class in its antecedent part, another round of the backward chaining procedure would be invoked, for each of these foreign classes forming a backward chain in the inference process.

The migrated rules, which would form cycle with the others in the backward chain of rules are excluded from the selection and the corresponding foreign class's values remain unknown. The rules referring to these unknown values become unfit and difficult to survive and hence the cyclic rules can be avoided.

After the backward chaining procedure fired all of the selected migrated rules, the local rule being considered would be evaluated, and a fitness value is assigned according to its classification accuracy.

2.3 Fitness Evaluation, Selection and Genetic Operators

The CRGP uses a support-confidence based fitness function and token competition to evaluate the local rules' fitness [1]. It employs three canonical genetic

Table 2. The pseudocode of the synthetic data set generation.

```
attribute 0 = random();
attribute 1 = random();
attribute 2 = random();
attribute 3 = random();
attribute 4 = random();
attribute 5 = random();
if (attribute 0 > attribute 1)
    class 0 = 1;
else
    class 0 = 0;
if (attribute 2 > attribute 3)
    class 1 = 1;
else
    class 1 = 0;
if (attribute 4 > attribute 5)
    class 2 = 1;
else
    class 2 = 0;
if (class 0 ≠ class 1)
    class 3 = 0;
else if (class 0 ≠ class 2)
    class 3 = 1;
else
    class 3 = 2;
if (class 0 ≠ class 1)
    class 4 = 0;
else if (class 0 ≠ class 2)
    class 4 = 1;
else
    class 4 = 2;
where :
 random() returns a real number between 0 and 999
```

operators for evolution; they are the crossover, mutation and dropping conditions [9]. It selects local rules by the Roulette Wheel method. Fitter local rules, higher chance to be selected. The selected local rules are applied with one of the genetic operators, to produce the offspring. Then, all the local rules and the offspring compete with each other. The best half of them would be selected as the new population for the next generation.

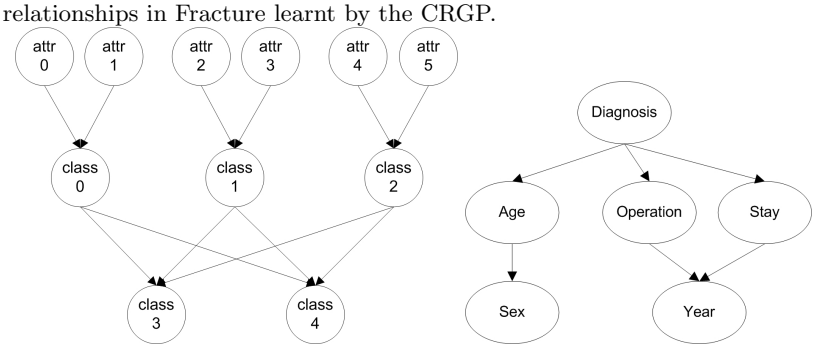
When the maximum number of epochs is met, the training process is terminated and all of the local rules of all the populations are collected together as the resultant rule set.

3 Experiments

The CRGP has been compared with the Multi-population Genetic Programming (MGP), C5.0, Belief Network PowerConstructor (PC), WinMine (WM) and B-Course (BC). The MGP is a canonical GP with multi-population. It has no migration, no backward chaining and all of the populations can access all of classes' values directly. We have implemented the CRGP and the MGP in C++. They have the same implementation details. The C5.0, PC, WM and BC are downloaded from their web sites [8, 4, 3, 2].

We have evaluated the CRGP on a synthetic and a real-life medical data set, a Fracture medical data set. The synthetic data set is a multi-class problem with 400 data items. Table 2 and figure 1(left) show the pseudocode used to generate the synthetic data set and the corresponding relationships among the attributes respectively. It has 6 input attributes, 5 classes and 400 data items. Classes 3 and 4 are generated by the same set of rules. They are used to evaluate if the algorithm can produce an acyclic rule set. The real-life data set, "Fracture", is from the Orthopaedie Department of the Prince of Wales Hospital of Hong Kong.

Fig. 1. (Left) The relationships among attributes in the synthetic data set. (Right) The relationships in Fracture learnt by the CRGP.



It consists of records of children with limb fractures admitted to the hospital in the period 1984-1996. "Fracture" has been used in [11]. It has 1 input attribute, 5 classes and 6574 data items.

The data sets are split into two parts. 66% of the data items are used for the training, the rest of them are used for the testing. For all of the algorithms, we have specified the input attributes. The values of the number of epochs, the maximum number of generations, the number of local rules in a population, the maximum depth, the crossover rate, the mutation rate and the dropping condition rate are 40, 10, 15, 0.5, 0.4 and 0.1 respectively.

3.1 Synthetic data set results

We construct the relationships learnt by the CRGP and MGP by, 1) evaluating the resultant rule set by the typical testing methodology of GP and removing the redundant rules [11]; and 2) collecting the remaining rules as the learnt relationships. Since GP is a stochastic algorithm, it produces slightly different result in each run and the results are likely to have some redundant codes. To remove the stochastic effect, we consider only the relationships exist in all of the runs as the actual results learnt.

Since the C5.0 can handle only one class at a time, in the experiment, it is executed several times, once for each class. The relationships learnt by the C5.0 are derived by combining the trees it learnt in each run.

The BN learning algorithms cannot handle continuous values directly. The continuous values are discretized first. The BC is a stochastic BN learning algorithm, we have evaluated it ten times and selected the best result.

Figure 2(left) shows the relationships learnt by the CRGP. The relationships are the same as the real relationships shown in figure 1(left). The additional relationship between classes 3 and 4 representing their values are generated by the same set of rules. The CRGP has learnt the relationships successfully. Figures 2(right), 3 and 4 show the relationships learnt by the MGP, PC, WM and BC

Fig. 2. (Left) The relationships learnt by the CRGP. (Right) The relationships learnt by the MGP.

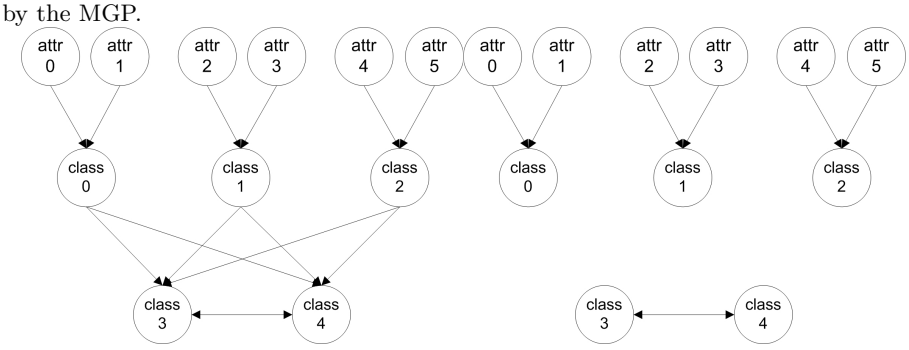
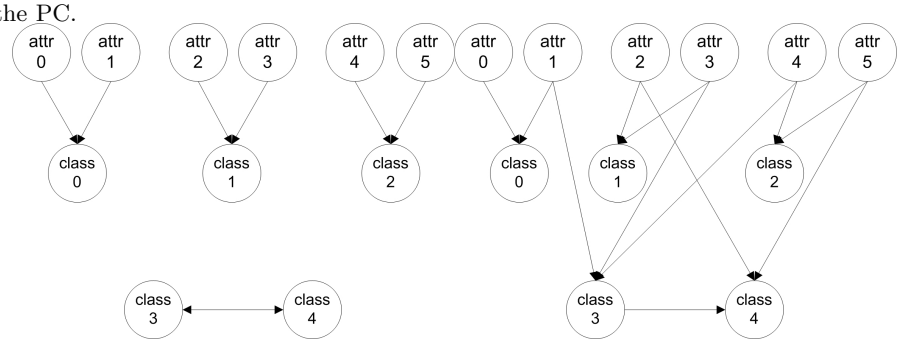


Fig. 3. (Left)The relationships learnt by the C5.0. (Right) The relationships learnt by the PC.



respectively. They show that only the BC can learn very similar relationships. The MGP, C5.0 and WM cannot learn the relationships among class 3, attributes 0, 1 and 2, and the one among class 4, attributes 0, 1 and 2. The PC cannot learn the relationship between classes 3 and 1, and the one between class 4 and 1; It has also learnt three incorrect relationships.

Tables 3(left) and 3(middle) illustrates the cyclic rules and trees produced by the MGP and C5.0 respectively. They show only the rules and trees with classes 3 and 4 of value 1. The rules and trees state that if class 3 is 1, then class 4 is 1, and vice versa. This is a cyclic phenomenon and we cannot infer anything for the classes given the input attributes' values.

Table 3(right) shows the corresponding rules produced by the CRGP, which are not cyclic. Besides the relationship between classes 3 and 4, the rules also represent inferences with classes 3 and 4 equaling to 1. The rules formed an acyclic inference chain and produces meaningful result. Also, the most import point is that the rules and the inference chains provide the relationships amongst

Fig. 4. (Left) The relationships learnt by the WM. (Right) The relationships learnt by the BC.

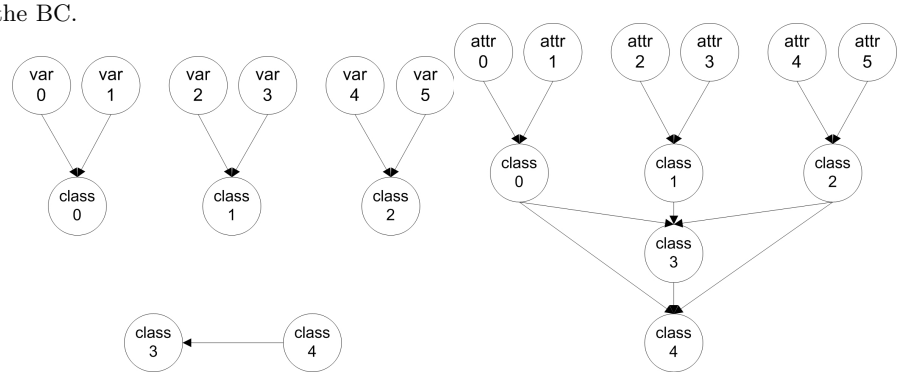


Table 3. (Left)The rules produced by the MGP. (Middle)The decision trees produced by the C5.0. (Right) The rules produced by the CRGP.

(= class4 1)→(= class3 1). (= class3 1)→(= class4 1). (= class4 1)→(= class3 1). (= class4 1)→(= class3 1). (= class4 1)→(= class3 1). (= class3 1)→(= class4 1). (= class3 1)→(= class4 1). (= class3 1)→(= class4 1).	Decision tree for Class 3: Class 4 = 0: 0 (122) Class 4 = 1: 1 (68) Class 4 = 2: 2 (74) Decision tree for Class 4: Class 3 = 0: 0 (129) Class 3 = 1: 1 (69) Class 3 = 2: 2 (66)	(∧ (≠ class2 class0)(= class1 class0))→(= class4 1). (= class4 1)→(= class3 1). (≠ class0 class2)→(= class3 1). (∧ (≠ class2 class0)(= class1 class0))→(= class4 1). (∧ (> attr2 attr3)(≠ class2 class0)→(= class4 1). (∧ (> attr2 attr3)(≠ class2 class0)→(= class4 1). (∧ (> attr2 attr3)(≠ class2 class0)→(= class4 1). (∧ (> attr2 attr3)(≠ class2 class0)→(= class4 1).

the inputs and class attributes. It is the advantage of using the backward chaining rather than inferring all of the classes' values directly from the given inputs.

The results show only the CRGP and BC can learn the relationships; the result of the MGP shows the multi-class problem cannot be solved by accessing all of classes' values directly; and the result of the C5.0 shows the multi-class problem cannot be solved by simply executing the algorithm several times.

3.2 Fracture results

Wong et al have learnt the relationships represented by BN in the Fracture data set [11]. The result is confirmed by the medical experts.

Figure 1(right) shows the relationships learnt by the CRGP. The network is exactly the same as the one learnt by Wong. It shows: 1) The Diagnosis affects the Operation and Stay, different fractures are treated with different operations and need different time for recovery; 2) The Diagnosis affects the Age. Some fractures occur more often in particular age groups; 3) There is a relationship between the Age and Sex. It is observed that the young patients are more likely to be female; and 4) The Operation and Stay affect the Year. It is observed that

the length of stay in hospital is different in different periods. The CRGP has learnt the relationships successfully.

4 Summary

We have described a novel algorithm, CRGP. It learns a set of classification rules and the relationships among attributes for multi-class problem. Unlike BN, it can handle input attributes with continuous values directly. We have evaluated its performance on a synthetic and a real-life medical data sets. The experimental results have shown the CRGP outperforms the MGP, C5.0, PC and WM; and it has the comparable result as the BC in relationship learning. With the flexibility of GP, we believe the CRGP can learn complex relationships amongst attributes with discrete or continuous values that cannot be handled by BN.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of otems in large databases. In *Proceedings of the 1993 International Conference on Management of Data*, pages 207–216, 1993.
2. Jie Cheng. Belief network powerconstructor. http://www.cs.ualberta.ca/ jcheng/bnpc.htm, 1998.
3. David Maxwell Chickering. The winmine toolkit. http://research.microsoft.com/ dmax/winmine/tooldoc.htm, 2002.
4. Helsinki Institute for Information Technology Complex Systems Computation Group. B-course. http://b-course.hiit.fi/, 2002.
5. Alex A. Freitas, editor. *Data Mining and Knowledge Discovery with Evolutionary Algorithms.* Springer, 2002.
6. N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. In *Machine Learning*, pages 131–163, 1998.
7. D. Heckerman and D. Geiger. Learning bayesian networks: a unification for discrete and gaussian domains. In *Proceedings of the eleventh conference on uncertainty in artificial intelligence*, pages 274–284, 1995.
8. RuleQuest Research Pty Ltd. Data mining tools see5 and c5.0. http://www.rulequest.com/see5-info.html, 2003.
9. R.S. Michalski. A theory and methodology of inductive learning. In J.G. Carbonell R.S. Michalski and T.M. Mitchell, editors, *Machine Learning - An Artificial Intelligence Approach*, chapter 4. Los Altos, Calif., 1983.
10. S. Monti and G. F. Cooper. Learning bayesian belief networks with neural network estimators. In *Advances in Neural Information Processing Systems*, pages 579–584, 1997.
11. Wong M. L. Lam W. Leung K. S. Ngan, P. S. and J. C. Y. Cheng. Medical data mining using evolutionary computation. In *Artificial Intelligent in Medicine, Special Issue On Data Mining Techniques and Applications in Medicine*, pages 73–96, 1999.
12. S.W. Wilson. Generalization in the xcs classifier system. In *Genetic Programming: Proceedings of the Third Annual Conference*, pages 665–674, 1998.
13. S.W. Wilson. Mining oblique data with xcs. In *International Workshop on Learning Classifier Systems*, page Extended Abstract, 2000.