# A Knowledge-Acquisition Strategy Based on Genetic Programming

Chan-Sheng Kuo[a], Tzung-Pei Hong[b, c], Chuen-Lung Chen[a]
[a]*Department of Management Information Systems*
*National Chengchi University, Taipei, Taiwan*
[b]*Department of Computer Science and Information Engineering*
*National University of Kaohsiung, Kaohsiung, Taiwan*
[c]*Department of Computer Science and Engineering*
*National Sun Yat-sen University, Kaohsiung, Taiwan*
*{cskuo@nccu.edu.tw, tphong@nuk.edu.tw, chencl@mis.nccu.edu.tw}*

## Abstract

*In this paper, we have modified our previous GP-based learning strategy to search for an appropriate classification tree. The proposed approach consists of three phases: knowledge creation, knowledge evolution, and knowledge output. One new genetic operator, separation, is designed in the proposed approach to remove contradiction, thus producing more accurate classification rules. A subtree pruning technique is also used to restrain the classification trees excessively expanding in the evolutionary process. Experimental results from diagnosis of breast cancers also show the feasibility of the proposed algorithm.*

## 1. Introduction

Knowledge management is used to assist enterprises for effective management of organizational knowledge. A knowledge management system (KMS) plays a role of information technique and needs the construction of a complete and consistent knowledge base for its effective application to support an organization [1]. Only a few knowledge workers will, however, contribute their knowledge into the knowledge repositories since they usually consider it as an extra burden [7]. Designing an effective method to automatically generate classification rules fed into knowledge bases thus plays a critical role in a knowledge management system.

Genetic programming is an evolutionary approach and has been used to discover some useful classification rules in medical domains [8]. In the past, we proposed a learning algorithm based on genetic programming to search for an appropriate classification tree [6]. In this paper, we will modify it and propose a GP-based knowledge-acquisition strategy to automatically find a good classification tree, which will be transferred into a rule set composed of some classification rules. The proposed approach adds a subtree pruning technique to restrain the classification trees excessively expanding in the evolutionary process. Besides, one new genetic operator, separation, is also designed in the proposed approach for removing contradiction among the rules.

## 2. Review of genetic programming

Genetic programming (GP), firstly proposed by Koza [4], used genetic operations as its basis for creating computer programs. The representation of an individual in GP is a tree structure consisting of functions and terminals.

There are three fundamental genetic operators, which are reproduction, crossover and mutation [5]. The reproduction operation selects a set of individuals from the population according to a selection method based on their fitness values. The individuals selected are reproduced into the new population in the next generation. Many selection methods have been proposed, which include fitness-proportionate reproduction, tournament selection and rank selection [4]. In this paper, the fitness-proportionate selection method is adopted.

Genetic programming has been applied to several applications such as economic models, medical applications, finance engineering, hand written digit recognition, classification, and among others [2][8].

0-7695-3038-9/07 $25.00 © 2007 IEEE
DOI 10.1109/ICCIT.2007.133

## 3. A GP-based knowledge acquisition strategy

In this section, we state the proposed knowledge-acquisition strategy based on genetic programming to generate a good classification tree, which can be transferred as a rule set and further fed into a knowledge base with reduced intervention of domain experts. The proposed approach is shown in Fig.1.

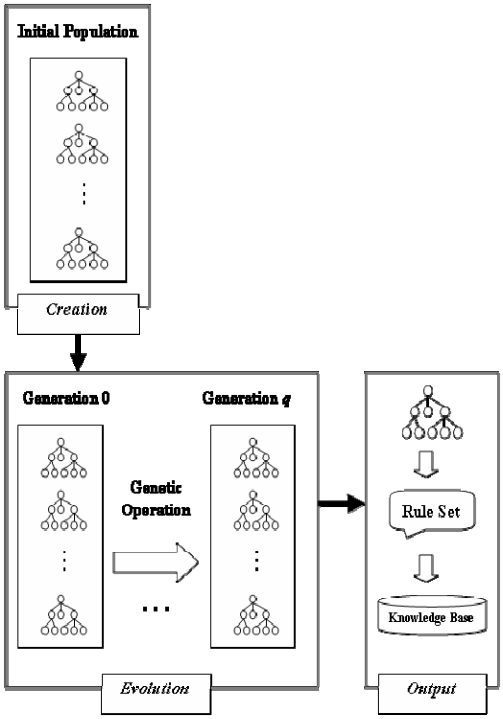


**Fig. 1. The three phases of the proposed approach.**

It consists of three phases: knowledge creation, knowledge evolution, and knowledge output. In the knowledge creation phase, *m* classification trees are randomly generated to form an initial knowledge population, which is then ready for evolution. In the knowledge evolution phase, each classification tree in the population is evaluated by a fitness function and a set of training instances to get the fitness value. The evolution phase then chooses suitable classification trees according to their fitness values to crossover, gradually producing good offspring trees and constructs new classification trees by the proposed genetic operations. The evolutionary process is repeated until a good classification tree is found. In the knowledge output phase, the final best classification tree is transferred as a rule set, and then output to one centralized knowledge base to facilitate decision making or daily operations in an organization.

## 4. The proposed GP algorithm for knowledge acquisition

The proposed GP algorithm for knowledge acquisition is stated in this section to show how an appropriate classification tree can be found. Its flowchart is shown in Fig. 2.

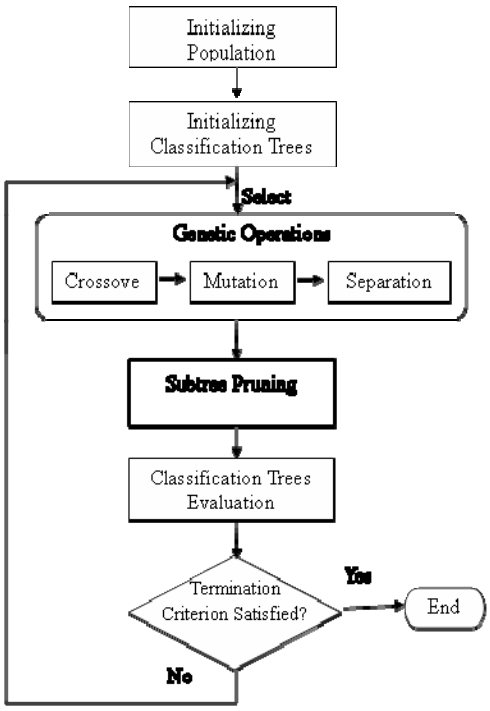


**Fig. 2. Flow chart of the proposed GP algorithm.**

Some details about the execution of the algorithm are stated below.

### A. The initial population

The technique of genetic programming requires a population of feasible solutions to be initialized and updated during the evolving process. Each individual within the population is a hierarchically structured tree consisting of functions and terminals. The initial population is randomly generated with some constraints for forming feasible classification trees. Below, a simple example is given to illustrate the concept.

***Example 1.*** Assume in breast cancer diagnosis, two classes {Non-cancer, Cancer} represented as (R {R0,

R1}), are to be distinguished by the six features {Mass Shape, Mass Margin, Mass Density, Calcification Size, Calcification Distribution, and Calcification Shape}. Assume the feature of Mass Shape has four possible values {round, oval, lobular, irregular} represented as (S {S1, S2, S3, S4}), the feature of Mass Margin has four possible values {circumscribed, obscured, indistinct, spiculated} represented as (M {M1, M2, M3, M4}), the feature of Mass Density has four possible values {high, equal, low, fat containing} represented as (D {D1, D2, D3, D4}), the feature of Calcification Size has five possible values {< 0.5 cm, 0.5 – 1 cm, 1 – 2 cm, 2 – 3 cm, > 3cm} represented as (I {I1, I2, I3, I4, I5}), the feature of Calcification Distribution has three possible values {grouped/clustered, regional, diffuse/ scattered} represented as (K {K1, K2, K3}), and the feature of Calcification Shape has four possible values {round or punctuated, coarse heterogeneous or irregular, fine pleomorphic, coarse or popcorn type} represented as (H {H1, H2, H3, H4}). To illustrate the tree representation more clearly, assume that a rule set $RS_i$ has the following only two rules:

$Rule_1$: If (Mass Shape = irregular) and (Mass Margin = speculated) and (Mass Density = high) then Class is Cancer;

$Rule_2$: If (Calcification size < 0.5 cm) and (Calcification Distribution = grouped/ clustered) and (Calcification Shape = round or punctuated) then Class is Non-cancer.

The two rules are equivalent to the following representation:

$Rule_1$': If S4 and M4 and D1 then R1;
$Rule_2$': If I1 and K1 and H1 then R0.

The two rules can be represented as a classification tree shown in Fig. 3.

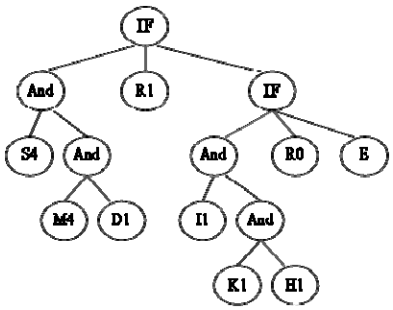


**Fig. 3. Representation of a classification tree.**

## B. The fitness function

The fitness function is used to evaluate how well an individual can be for solving the given problem in the evolutionary process. In order to develop a good knowledge base from an initial population of classification trees, GP selects parent classification trees with high fitness values for mating. Two important factors, accuracy and complexity, are used in evaluating a classification tree [9]. The accuracy for a classification tree *CT* is evaluated by the training instances and is defined as follows:

$$\text{Accuracy}(CT) = \frac{\text{the total number of training instances correctly matched by CT}}{\text{the total number of training instances}}$$

The complexity of a classification tree *CT* is the ratio of nodes used, which is defined as follows:

$$\text{Complexity}(CT) = \frac{\text{number of nodes within CT}}{\left[\sum_{i=1}^{j}(\text{number of nodes within initial CT}_i)\right] \Big/ j}$$

where *j* is the number of individuals in a population. Accuracy and complexity are combined to represent the fitness value of a classification tree. The evaluation function is thus defined as follows:

$$\text{fitness}(CT) = \frac{[\text{Accuracy}(CT)]}{[\text{Complexity}(CT)]^{\alpha}}$$

where $\alpha$ is a control parameter representing the tradeoff between accuracy and complexity.

## C. The subtree pruning

In our approach, the subtree pruning is used to avoid the classification trees excessively expanding and reduce misclassification. It removes the rules which have the confidence value below the predefined confidence threshold in a classification tree. The confidence value can represent the degree of certainty for each rule and is between 0 and 1. As the confidence value of a rule approaches to 1, it means the rule is highly important and valuable.

Let the classification tree to be processed be *CT* with *p* rules. Assume $Rule_i$ is included in the classification tree *CT*. The confidence value $c_i$ of the rule $Rule_i$ is evaluated using the test instances as follows:

$$c_i = t_i / m_i,$$

where $m_i$ is the numbers of test instances correctly matched with the antecedent of $Rule_i$ and $t_i$ is the numbers of test instances correctly matched with both the antecedent and the consequent of $Rule_i$.

The steps for the subtree pruning are described as follows.

(1) Set $h = 1$, where $h$ is used to represent the index of the selected rule in a classification tree $CT$ for checking.
(2) Set the confidence threshold as $\lambda$.
(3) Calculate the confidence value $c_h$ of $Rule_h$.
(4) Check whether $c_h$ is lower than the confidence threshold $\lambda$. If $c_h$ satisfies the above condition, remove $Rule_h$ from the classification tree $CT$.
(5) Set $h = h + 1$.
(6) If $h > p$, then exit the searching process; otherwise, repeat Steps 3 to 6.

***Example 2***: Assume a classification tree $CT$ shown in the left side of Fig. 4 includes the three rules $Rule_1$, $Rule_2$, and $Rule_3$. Also assume the confidence values of the three rules are first calculated as $c_1$ =75%, $c_2$ = 19%, and $c_3$ = 63% from the set of test cases. The confidence threshold $\lambda$ is set at 25%. By the subtree pruning, $Rule_2$ is thus removed from the classification tree $CT$ since its confidence value (19%) is smaller than the confidence threshold (25%). The resulting classification tree $CT'$ is then shown at the right side of Fig. 4.

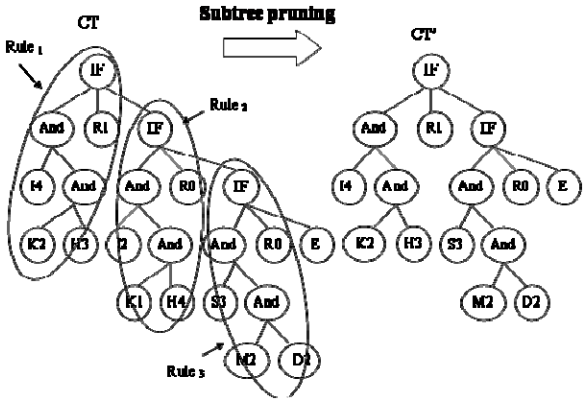


**Fig. 4. An example of subtree pruning.**

## D. Genetic operators

In this paper, two fundamental genetic operators, crossover and mutation, and one new operator, separation, is used in the proposed algorithm. The separation operator is designed to solve the contradiction problem [3].

***1) Crossover:*** In the crossover operation, two parent trees are partially exchanged to form two new offspring trees. The crossover point may occur within a rule or on rule boundaries.

***2) Mutation:*** The mutation operation begins by selecting a parent tree from the population at random. A node within the tree is randomly selected. The mutation operation then replaces the selected node (including its child nodes) with a randomly generated subtree in its place.

***3) Separation:*** The separation operation solves the contradiction problem, in which two rules with the same feature values conclude to different classes. It removes some rules from a contradictive classification tree, such that the tree can become consistent. It can thus produce more consistent classification rules and promote the rule accuracy. Let the classification tree to be processed be $CT$ with $k$ rules. The steps for the separation operation are shown as follows.

(1) Set $i = 1$ and $j = i + 1$, where $i$ and $j$ are used to represent the indexes of the current two selected rules for checking.
(2) Compare $Rule_i$ and $Rule_j$ in $CT$ for their feature values and classes.
(3) If the two rules have the same feature values but different classes, split $CT$ into two subtrees, $CT_1$ and $CT_2$, to respectively exclude a contradictive rule ($Rule_i$ or $Rule_j$) from $CT$, calculate their accuracy values, and select the subtree with the higher accuracy value as the individual.
(4) Set $j = j + 1$.
(5) If $j > k$, then do the next step; otherwise, go to step (2).
(6) Set $j = i + 2$ and $i = i + 1$.
(7) If $i = k$, then exit the searching process; otherwise, go to step (2).

# 5. Experimental results

In this section, a set of data for breast cancer diagnosis was used to test the performance of the proposed approach. There were 372 cases in the experiments obtained from a hospital in Taipei, Taiwan. The cases consisted of thirteen features and two classes (Cancer and Non-cancer). The goal was to find an integrated classification tree which could be converted into a set of rules to help identify one of the two classes.

In the experiments, the rates for crossover, mutation, and separation operations were set at 0.9, 0.02, and 0.8, respectively. The confidence threshold was set at 0.01. Moreover, the parameter $\alpha$ in the fitness function was used as a tradeoff between accuracy and complexity, and was set at 0.125. The error rates of the proposed approach and the traditional GP method were compared. Note that the latter method didn't use subtree pruning and separation operation. The proposed approach obtained an error rate of 21.65% and the traditional GP method obtained an error rate of 23.47% after 500 generations, averaged over 5 runs. Experimental results thus showed that the proposed approach had a lower error rate than the traditional GP method.

## 6. Conclusions

In this paper, we have proposed a knowledge-acquisition approach based on the technique of genetic programming to automatically search for an appropriate classification tree according to the criteria of accuracy and complexity. It tries to find a good classification tree by genetic operators and improves its accuracy via the fitness function. Experimental results have shown that the proposed approach could find a good classification tree with a lower error rate than the traditional GP method.

The proposed approach has designed the subtree pruning to avoid the classification trees excessively expanding and the separation operator to take domain-specific characteristics into consideration, thus getting results closer to those desired. It, however, takes more execution time than using only the original operators. In the future, we will study the choice of fitness functions and additional operators in order to further reduce error rates.

## Acknowledgements

This research was supported by the National Science Council of the Republic of China under contract NSC 95-2221-E-390-025.

## References

[1] M. Alavi, and D. E. Leidner, "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues", *MIS Quarterly*, vol. 25, no. 1, 2001, pp. 107 - 136.

[2] S. H. Chen and T. W. Kuo, "Evolutionary Computation in Economics and Finance: A Bibliography", *Evolutionary Computation in Economics and Finance*, Physica-Verlag, Heidelberg New York, 2002, pp. 419-455.

[3] J. Giarratano and G. Riley, *Expert System Principles and Programming*. Boston, MA: PWS, 1993.

[4] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.

[5] J. R. Koza, *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann, 1999.

[6] C. S. Kuo, T. P. Hong and C. L. Chen, "Learning Classification Trees by Genetic Programming," *The 2006 International Conference on Hybrid Information Technology*, Korea, 2006.

[7] M. M. Kwan and P. Balasubramanian, "Knowledge Scope: Managing Knowledge in Context", *Decision Support Systems*, vol. 35, 2003, pp. 467-486.

[8] A. Tsakonas, G. Dounias, J. Jantzen, H. Axer, B. Bjerregaard and D. G. Keyserlingk, "Evolving Rule_ Based Systems in Two Medical Domains Using Genetic Programming", *Artificial Intelligence in Medicine*, vol. 32, 2004, pp. 195-216.

[9] 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*, vol. 28, no. 3, 1998, pp.471-476.